ERIC Document Resume | See Instruction Sheet | DATE OF RESUME | DATE RECEIVED

A. PUBLICATION DATA

1. ERIC NUMBER ED001255

2. ORIGINATOR CONTROL NO(S). NA

3. IDENTIFICATION NO. CR 1323

DO NOT WRITE IN SPACE

MICROFILM COPY AVAILABLE YES ☒ NO ☐

COPYRIGHT GRANTED YES ☐ NO ☒

COPYRIGHT RELEASE FILED YES ☐ NO ☒

TYPE OF REPORT
PROPOSAL ☐
IN-PROGRESS ☐
FINAL ☒
FOLLOW-UP ☐

OTHER ERIC REFERENCES:

ERIC SATELLITE CODE

4. AUTHOR(S): Holzman, A.G. and Turkes, W.R.

5. TITLE OF DOCUMENT ABSTRACTED: Optimal Scheduling in Educational Institutions

6. PUBLICATION TITLE: NA

7. EDITOR(S): NA

8. PLACE OF ORIGIN
ORG. Univ. of Pittsburgh
ST. Pittsburgh, Penna.
CITY, STATE
TELE. EXT.
INST/LOC CODE

9. PUBLISHER OR SOURCE Cooperative Research Program, Off. of Education, Dept. of HEW, Washington, D. C.

10. VOL. NA | 11. PAGES NA | 12. MO./YR. 64 | 13. TOTAL PP. 453

B. INDEXING DATA

14. SUBJECT AREA

15. TYPE OF ENDEAVOR Research

16. INDEXING TERMS (6-12)

		Parameters	Time blocks	
Course offerings	Linear programming	Plant Availability		
Course requests	Max. requirements	Schedule conflicts		
Faculty availabil.	Min. requirements	Schedule cycles		

17. ABSTRACT (MAX. 250 WORDS)

This research project was initiated to develop, by computer, a master program for high schools and universities in order to free administrators from the details of this burdensome task. The research was separated from preconceived notions of educational scheduling systems and the generation of a master schedule by computer was emphasized. The generated master schedule is contingent upon specific course schedules. Feedback between the scheduling system and a student's master program is essential. To make a program computationally feasible, further research on the probabilistic aggregate approach is indicated. Sample input and output data, plus mathematical formulae and tables are included in text.

ERIC No.

18. SOURCE OF ABSTRACT

C. FUNDING DATA

19. SUPPORTING ORGANIZATION

20. DIVISION OR DEPT.

21. CONTRACT NO.

PROJECT NO.

22. TOTAL COST OF PROJECT

23. DURATION OR PROJECT

U.S. Department of Health, Education, and Welfare
Office of Education

REPRINTED BY PERMISSION
U. S. DEPARTMENT OF HEALTH, EDUCATION AND WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN PRINTED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

ED001255

OPTIMAL SCHEDULING IN EDUCATIONAL INSTITUTIONS

Cooperative Research Project No. 1323

A. G. HOLZMAN
W. R. TURKES

University of Pittsburgh

BLANK PAGE

OPTIMAL SCHEDULING
IN
EDUCATIONAL INSTITUTIONS

Cooperative Research Project No. 1323

A. G. Holzman
W. R. Turkes

University of Pittsburgh
Pittsburgh, Pennsylvania
1964

The research reported herein was supported by the Cooperative Research Program of the Office of Education, U.S. Department of Health, Education, and Welfare.

TABLE OF CONTENTS

Page

1. PROBLEM ON WHICH THE RESEARCH WAS FOCUSED 1

2. OBJECTIVES AND/OR HYPOTHESES 10

3. RELATED RESEARCH 12

Introduction 13
Assignment (Sectioning) Programs 14
Master Schedule Construction 19

4. PROCEDURE, ANALYSES OF DATA, AND FINDINGS 28

ALGORITHMIC SCHEDULER 29

BOOLEAN ISOMORPHIC STRUCTURES - A.G. Holzman and I.B. Turksen 30
Model 31
Behavior Of The Boolean Structure . . . 37
Algorithms 47
Summary 60

RELATIONS MATRIX STRUCTURES - I.B. Turksen and A.G. Holzman 62
New Concepts 65
Interpretation of Relations Matrix . . . 68
A Policy Procedure 75
Summary 80

DECOMPOSED MODEL OF AN EDUCATIONAL SCHEDULING SYSTEM FOR OPTIMAL MASTER SCHEDULE CYCLE - I.B. Turksen and A.G. Holzman 81
Basic Concepts 82
Phase I A. Determination Of Course Sections To Be Offered 85
Phase I B. Administrative Changes . . . 91
Phase II. Assignment Of Course Sections To Faculty, Students, And Physical Plant 96
Phase III. Development Of The Constraint Structures Imposed On The System 100
Phase IV. Establishment Of Priorities Based On The System Constraints And The Course Specifications 111
Phase V. Construction Of The Master Schedule To Satisfy The Conditions Established In Phases I Through IV 119
Summary 150

Page

- CONTROLLED SCHEDULE CYCLE -
  - I.B. Turksen and A.G. Holzman 153
    - Introduction 154
    - Criteria Revisions 155
    - Controlled Schedule Cycle 156
    - Extension To The Algorithm 160
    - Summary 161

- DOCUMENTATION OF MASTER SCHEDULE PROGRAMS -
  - I.B. Turksen and A.G. Holzman 163
    - Program Of Optimal Master Schedule Cycle 164
    - Program of Controlled Schedule Cycle 167

- SUGGESTED EXTENSIONS IN THE CONTROLLED SCHEDULE CYCLE CONCEPT -
  - I.B. Turksen and A.G. Holzman 236
    - Further Research 237
    - Improvement Of The Criteria Which Are Used To Minimize The Number Of Conflicts 239
    - Further Implications Of Conditional Probability Concept 242

LINEAR PROGRAMMING SCHEDULER 243

- THE LINEAR PROGRAMMING APPROACH TO MASTER SCHEDULE GENERATION -
  - R.E. Harding 244
    - Introduction 245
    - The General Linear Model 246
    - The Turtle Creek Senior High School Example 269
    - Advantages Of L.P. Approach 277
    - Research Needs 281
    - Glossary 294

HEURISTIC SCHEDULER 297

- A HEURISTIC APPROACH TO THE GENERATION OF A MASTER SCHEDULE AND ASSIGNMENT OF STUDENTS -
  - C.E. Donaghey and O.S. Ozkul 298
    - Introduction 299
    - Initial Work 299
    - Initial Results With University Data 308
    - Results With High School Data 317
    - Revised Master Schedule Generator . . . 321

Page

An Improved Assignment Technique 323
Further Research 328

5. CONCLUSIONS AND IMPLICATIONS 330

6. APPENDIX 336

PROBABILISTIC AGGREGATE APPROACH - P.J. Kiviat 337

ASSIGNMENT OF STUDENTS TO COURSE SECTIONS IN AN EDUCATIONAL SCHEDULING PROBLEM - O.S. Ozkul 352

ROUTINE TO GENERATE COURSE REQUESTS AND PROVIDE FILE MAINTENANCE FOR STUDENT RECORDS - W.E. Baughman 386

BIBLIOGRAPHY-ABSTRACTS - P.M. Dargay 427

LIST OF COMPUTER PROGRAMS

Page

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE -
I.B. Turksen 168

PROGRAM OF CONTROLLED SCHEDULE CYCLE -
I.B. Turksen 206

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO
COURSE SECTIONS -
O.S. Ozkul 368

PROGRAM TO GENERATE COURSE REQUESTS AND
PROVIDE FILE MAINTENANCE FOR STUDENT RECORDS -
W.E. Baughman 402

1. PROBLEM ON WHICH THE RESEARCH WAS FOCUSED

INTRODUCTION

The system in educational institutions for scheduling students, assigning faculty, and utilization of facilities has become extremely inadequate to meet the present demands. With the enrollment in high schools and colleges spiralling and with the trend toward additional terms per year, the frequency and the complexity of these activities have been increased, and the problem has become more acute. It is unfortunate that the faculty and the administrators of secondary schools and colleges be burdened with the task of performing these detailed, yet essential functions term after term, year after year. Their time is needed so urgently for professional counselling, guidance, and instruction of the student; for concentrated effort on research in carving out new frontiers of knowledge; for preparation of publications; and for a continuous evaluation of the educational system to shape the policies necessary in the quest for excellence in the educational process.

The growth of educational institutions have spawned many new complexes in the scheduling system. Thus the need for focusing on a more effective scheduling system relative to these new complexes has become more urgent.

At the University of Pittsburgh the research has been divorced from the pre-conceived notions of educational scheduling systems. Basic questions, such as the following, have been investigated.

What is the nature of this system?

What are its inherent characteristics?

BLANK PAGE

What direction does it take?

What, if any, are the natural end-points of the process?

The research has not been limited to one particular approach, but instead, a comprehensive investigation, analysis and design of the educational scheduling system was considered to be the most valuable orientation for the project. The major efforts were devoted to the following three prong attack on the scheduling system, with emphasis on the generation of master schedules.

1. Algorithmic Scheduler
2. The Linear Programming Scheduler
3. Heuristic Scheduler

Additional research, though not as exhaustive, has been conducted on a Probabilistic Aggregate Approach to Master Schedule Construction, an Assignment of Students to Course Sections in an Educational Scheduling Model, and a Routine to Generate Course Requests and Provide File Maintenance for Student Records.

Throughout the many facets of this project, an attempt was made not to limit the results of the research to a particular school, but to generalize when possible, to provide for a more universal application of the findings.

The Algorithmic Scheduler starts with Boolean Isomorphic Structures in an Educational Scheduling Model. The model of a general class scheduling system is developed, and its environment is explored. A boolean basis tableau approach is introduced. Then, the cyclic behavior of the subsets of a given constraint and the isomorphic correspondence between the binary configurations and the positive integers lead to the development of general algorithms.

A methodology is developed to generate a feasible solution space which does not violate any of the constraints of the problem. It was found that the feasible solution space contained many possible combinations for time assignments according to some criterion. The research then focuses on the criterion to minimize the number of time periods required to satisfy all the constraints imposed upon the scheduling system. A Relations Matrix is constructed from the constraints of the system. Selection of the minimum number of time period vectors is obtained with the specifications of a policy and Bellman's principle of optimality.

The Relations Matrix Structures provide a basic framework for the decomposition of an educational scheduling system model so that the optimal master schedule cycle can be determined. The Decomposed Model of an Educational Scheduling System for Optimal Master Schedule Cycle consists of the following five phases:

Phase I. A. Determination of Course Sections to be Offered
B. Administrative Changes

Phase II. Assignment of Course Sections to Faculty, Students, and Physical Plant

Phase III. Development of the Constraint Structures Imposed on the System

Phase IV. Establishment of Priorities Based on the System Constraints

Phase V. Construction of the Master Schedule to Satisfy the Conditions Established in Phase I through IV.

Certainly it is possible that the number of periods required to satisfy all the constraints on the scheduling system may exceed the number of periods permitted for a cycle. In fact, the experience of the research team indicates that this is usually true in realistic scheduling problems. Thus, if the non-conflicting schedule cycle exceeds the maximum number of periods permitted in a cycle, then the constraints of the system must be relaxed. Relaxation resulted in the following revised criteria.

1. Minimum number of conflicts permitted
2. Master schedule must not require more than a specified number of days in a schedule cycle.

This is referred to as a Controlled Schedule Cycle.

A second prong in this research project is the Linear Programming Approach to Master Time Schedule Generation in Education. A general linear mathematical model of the master schedule restraining relations is constructed, and initial steps are taken to make the large scale system so generated an operational one.

This technique was applied to a real high school scheduling problem to ascertain the practical problems involved. The computer solution using the linear programming technique is compared with the manually prepared solution.

The linear programming approach taken in this project is a unique attack on the educational scheduling problem. Other people who have attempted this method abandoned it when they visualized the size of the model with its attendant problems, rather than taking a direct frontal attack on the issues that size creates.

While considerable work has been done in this project on the solution of a scheduling problem by linear programming, extensive research is required to develop the potential of this approach.

A linear programming solution of the system does provide useful by-product information, which administrators can utilize effectively. One of the most beneficial types of information is the so-called shadow price, which enables the administrator to evaluate the marginal returns of potential changes in physical plant, personnel resources, student demand, and administrative policies.

The third major prong of this project is A Heuristic Approach to the Generation of a Master Schedule and Assignment of Students. The initial thinking on this approach was that if a master schedule with no faculty or room conflicts could be developed, then, by scheduling all students to this master schedule, determine the course that caused the greatest number of conflicts, rescheduling this course, and repeating the cycle, a master schedule would be obtained in which the student conflicts were reduced to a minimum.

A computer program was developed using this approach, and was tested with both university and high school scheduling data. An analysis of these tests indicated that the results were satisfactory for problems that were not too restrictive. Orientation of the research was then directed to improving the program for the more restrictive cases. A revised master schedule generator was then constructed, with student course requests considered in this new program. Experimentation was again conducted, which pointed-out the desirability of more flexible time patterns.

An improved assignment technique is developed to go back two levels of unscheduling of students' course requests in an attempt to obtain a full schedule for a student.

As mentioned previously, additional research, though not as thoroughly explored as the major prongs of the project, was conducted in three other areas. The Probabilistic Aggregate Approach is one of combining student course demand for the creation of a master schedule to which individual students are assigned. The procedure is to generate schedules based on a-priori estimates of student demand. Individual students are then assigned by means of an iterative assignment algorithm which converges to a best overall student assignment. Thus far a theoretical investigation of this approach has been made. A computer program must be developed for investigating properties of the assignment procedure. The effect of the schedule density on the time required for schedule generation, and the convergence properties of the assignment algorithm should be investigated.

Another auxiliary area considered in this project is the Assignment of Students to Course Sections in an Educational Scheduling Problem. This is a very critical phase of the scheduling process, since the assignment of a student to a specific course section of a multi-section course is reflected in the construction of the master schedule. Thus, the master schedule is dependent upon the assignment of the specific course sections. These assignments are made on the basis of various criteria.

Computer programs have been written in this research project to assign students according to any one of the following criteria.

1. Assignment of students to course sections in accordance with their faculty preference.
2. Balancing the sections with respect to the number of students.
3. Homogeneous grouping of the students within the sections.
4. Hetereogeneous balancing of the sections with respect to the different groups in the student body.
5. Combination of homogeneous grouping with the criterion of balancing.
6. Combination of homogeneous grouping with the criteria of balancing.

These criteria are in effect established by administrative policy. The scheduler would probably desire a criterion which minimizes the number of constraints generated by the student assignments to sections. Ideas have been conceived on use of such a criterion, but no operational program has been developed.

The last area developed in this project was the Routine to Generate Course Requests and Provide File Maintenance for Student Records. This program is designed basically to have the computer determine the student course requests as opposed to the conventional method of having the student and/or his adviser select the courses. Criteria used by the student and adviser in the determination of courses to be taken are built into the computer program. An auxiliary function of the system is to maintain and update student records and point grade reports.

2. OBJECTIVES AND/OR HYPOTHESES

The objectives and/or hypotheses for this research project are:

1. To divorce thinking from a piecemeal modification of the present scheduling systems, and to concentrate on the development of new systems which will enable academic institutions to approach optimum solutions to the dynamic scheduling and attendant planning functions.
2. To endeavor to develop a scheduling system for students, faculty, and facilities which will have a widespread application in all academic institutions, universities, colleges, and high schools.
3. To develop mathematical models which adhere to criteria and logic, inherent in the generation of time schedules and in the assignment of students and resources.
4. To measure the effectiveness of the scheduling system in terms of applicable criteria.
5. To postulate scheduling systems which will of necessity force an objective, searching analysis, thus reducing the subjectivity and individual inconsistencies which presently jeopardize the systems used.

3. RELATED RESEARCH

INTRODUCTION

Educational scheduling systems have been studied by many groups during the last few years.* The realization of the objectives of these groups will tend to automate, at least to some degree, the scheduling functions. However, the approaches and methodology employed have varied in scope and depth. Intensive research and analyses have exposed hidden peculiarities of the scheduling process, concomitant with shedding light on scheduling phenomena. Some understanding of this complex function is beginning to emerge as a result of the research conducted.

In fact, some authorities have felt confident enough to categorize the activities of class scheduling into two disjoint processes. The most common categorization is: (1) construction of the master schedule, which is often referred to as a "time-table," and (2) assignment of students to the sections of courses for which time periods are indicated in the master schedule. This is often called "sectioning".

This categorization is somewhat a carry-over from the earlier concepts of educational scheduling. When the pioneering concepts of automated class scheduling were introduced, it was recognized that the construction of a master schedule was not as tractable to automatizing as was "sectioning". Thus the early efforts were directed to the assignment phase with the general notion that it is better to construct master schedules manually, based on past experience and intuitive judgment.

---

*The Appendix contains Biographical Abstracts of selected sources.

ASSIGNMENT (SECTIONING) PROGRAMS

Thus, the "sectioning" phase was developed with the assumption that a master schedule exists. The development of the assignment phase was initiated at the University of Purdue, and later on continued by others, with significant contributions having been made at IBM, Harvard University, and the Service Bureau Corporation. Among the universities actually using assignment programs in their registration activities are the University of Purdue, the University of Massachusetts, and the University of Illinois. High schools are also beginning to utilize the assignment programs.

Although no two "sectioning" programs are the same, nevertheless, the core characteristics of most programs do have similarities, as depicted in Figure 1. It is not meant to imply that all programs pattern exactly the characteristics exhibited in the figure, since idiosyncrasies exist for each.

CORE CHARACTERISTICS OF ASSIGNMENT PROGRAMS

1. Pre-assignment preparation
    a. Tally of approved course requests
    b. Analysis of student demand matrix
    c. Manual preparation of master schedule
2. Computer Input
    a. Master Schedule
        1. Course section cards - arranged in order of scheduling difficulty
            a. Section identification
            b. Seats and meeting times
        2. Study hall cards - only for high schools
    b. Student request cards - arranged in order of scheduling difficulty
        1. Student identification
        2. Course requests
3. Assignment of students to sections based on selected criteria
    a. Balanced sections
    b. Sequential assignment
    c. Student course requests arranged with course having least number of sections first
    d. Backtracking to un-schedule previously scheduled courses in order to resolve conflicts
4. Computer Output
    a. Master schedule
    b. Scheduled students
    c. Un-scheduled students
    d. Number of seats remaining for each section

Figure 1

The first operation is the tallying of student course requests. The computer corporations offer as a service the depicting of the students' requests in a student demand matrix which shows the number of students requesting each course and the number of students having joint-requests for each pair of courses. School administrators analyze the matrix as an aid in avoiding conflicting situations when preparing the master schedule of classes.

Before generating the student demand matrix, the Service Bureau Corporation has the school authorities assign a priority number to the courses being offered. These priority numbers express the relative degree of scheduling difficulty for each course, and enable the cross-referencing of the more-difficult-to-schedule courses with other difficult-to-schedule courses in the student demand matrix. A maximum of 200 courses can be handled with 60 of these courses having a priority code below a pre-set maximum priority number.

The master schedule and student request cards are inputs for the assignment process. The master schedule course cards are arranged with the most difficult-to-schedule courses and sections first. Along with data information, each course card indicates the time periods and days that the section meets and the number of seats available for each class.

In the program of G.E. Anderson from Harvard University, the course section cards specify a minimum and a maximum number of available seats.

Multiple time period classes and courses which do not meet everyday of the week can be scheduled. Dummy courses - lunch, free time, work, or dismissed - are created to handle special requests. Honor sections are designated as separate courses.

The assignment packages for high schools include as inputs a list of the study halls to be offered. An input card is required for each possible time pattern that will fill-in a student's class schedule.

Student request cards are arranged in order of scheduling difficulty as determined by the school administrators, or by some priority listing, such as seniors first, juniors next, etc. The student's input card contains information about the student and his course requests.

Anderson's student input cards can also indicate artificial course requests. These are requests for non-essential courses which can be replaced by approved substitutes. Each student also has a house number, or class level or standing on Anderson's input card.

Lunch can be handled as a course request. The Service Bureau Corporation permits the student to have a choice of several lunch time periods. If no preference is indicated, lunch will be assigned by the computer.

An added feature of the IBM assignment is the avoidance of a number of sequential study halls - if possible. This limit is determined by school administrators.

Scheduling of more than one term at the same time is permitted with the assignment programs. Of course, each term's master schedule and student request cards must be included as inputs. Limitations are imposed on parameters of the master schedule, such as the total number of course sections, number of student requests, and the number of time periods per day. For example, the IBM CLASS program can handle 500 course sections, the Service Bureau Corporation's program can accommodate 600 course sections, and Anderson's program can accept 1200 course sections.

The external inputs are translated into machine codes by the computers. All sections of a course are interchangeable. This permits a reduction in possible conflicts and a balancing of class sizes for the sections of a course. Now students are assigned sequentially.

The student's subject requests are ordered with the course having the least number of sections being scheduled first. The section with the most remaining seats is selected and assigned if possible. If not, the section having the next most seats is tried, and so on until a schedulable section is found. Then the next course request is scheduled. If all sections of a course are attempted and a conflict exists, the computer backtracks to the last scheduled course and re-schedules another of its sections before proceeding. An irresolvable conflict arises when all possible combinations of the sections are tried and no schedule is possible. Study halls are assigned to the un-scheduled times of the high school student whose course requests are satisfied.

Anderson's assignment procedures vary from the others. Course requests are arranged in ascending order. A student's house is matched with the house of the section of the requested course. If not possible, the computer selects the section with the most seats remaining. A minimum mode of search is initially attempted. If all possible combinations of sections yield no schedule, the maximum mode is tried. This maximum mode of search may open some sections because the maximum number of seats usually exceeds the minimum number which was specified on the input card for each section.

The output of the assignment programs include the master schedule, schedules of assigned students, and a listing of the rejected students. The up-dated master schedule also designates the number of unassigned seats for each section. Various messages can also be printed about the conflicts - course request not on master schedule, section closed, all combinations tried, etc.. Using the various outputs, the school administrators then manually resolve the conflicts.

MASTER SCHEDULE CONSTRUCTION

The early concepts have changed recently, as a series of feasibility studies have been conducted in an effort to obtain deeper insights in the construction of master schedules.

A heuristic simulation approach to master schedule construction has been developed at MIT, under the direction of R.E. Holz. The program is called GASP (General Academic Simulation Program). Holz states, "The program can be thought of as a tool to speed the production of schedules, and to speed the evolution of scheduling possibilities in a given school. The program will not take over the job of scheduling, but will be a good tool to help the scheduler do his job better. Important scheduling decisions remain with the key faculty and administrative personnel responsible for the schedule." The approach used is to make 10 to 20 computer runs, spaced a day or two apart, and thus obtain better and better approximations to a required schedule.

At Stanford considerable research has been done on mechanization of a variable high school schedule. This heuristic work reflects the thinking of D.W. Allen, J. Caffrey, R.V. Oakford, and C.R. Pack, on the development of flexible scheduling in high schools. The terms "flexible" or "variable" refer to class meetings of different lengths and frequency. Master schedules for high schools have been developed, with unresolved conflicts eliminated by manual modification of students' requests.

C.C. Gotlieb of the University of Toronto has taken a mathematical approach to the construction of Class-Teacher Time-Tables. Due to the complexity of the University master schedule, Gotlieb has directed his effort to a rigorous mathematical development for determining time assignments only for teachers to classes. Gotlieb's iteration procedure starts with certain initial conditions; the time-table has been partially filled for the critical situations, i.e. lunch periods, meetings, critical course assignments. Application of this mathematical procedure to large scale, practical problems has not as yet been reported by Gotlieb.

G.H. Sherman developed a mathematical approach to the scheduling problem for his doctoral dissertation at Purdue. The title of his work is Combinatorial Scheduling: On Finding a Partition of a Finite Set Which Maximizes a Set Function.

First, a brief review of the heuristic simulations of Holz and Pack will be presented. Then a compendium of the mathematical approaches of Gotlieb and Sherman will follow.

HEURISTIC SIMULATORS

The input for each course section includes the time pattern which indicates if the section meets for multiple time periods, the number of days per week, or the specific days of the week. Pack defines a course by structure type, which presets the number of consecutive time periods a section meets. Section input arrangement is in the order of desired scheduling as the programs schedule each class sequentially.

Teachers are designated for the subjects which they are qualified to instruct. Teachers can have preferences for certain sections or can be assigned to instruct certain sections. Pack has the school administrators indicate first and second choices of teachers for courses.

Special features of the rooms are specified as input. Rooms may be restricted to a specific course or a group of courses.

Holz uses a sample of the students' requests as input for the development of the master schedule. In Holz's programs, staff, space, and students are assigned a weight factor by the school authorities. These weights are used in the selection of the times for the course to meet. Student request cards are arranged in the order of their preference for the sequence of assignments.

The input is converted by the computer into binary vectors or matrices. Time vectors indicate availability of rooms, teachers, and students. A student also has a vector generated which indicates the particular courses he requests. Room and teacher vectors or matrices are also generated to indicate courses to which they can be assigned.

The programs developed by Holz construct the entire master schedule before student assignment. After the matrices have been formed, the sequential assignment of classes to times begins. By manipulation of staff, space, and sample student matrices, combinations of available times are found that satisfy the time pattern of that particular course section. Selection of the times to which the section is scheduled is made on the basis of the weights applied to the resource inputs. Instructor, room, and sample students are then assigned to that class, and then their availability matrices are up-dated. The next section offering is scheduled in a similar manner until the master schedule is completed.

Student assignment is sequential. Holz's program orders a student's course requests with the course having the least number of sections being arranged first. A conflict matrix is generated to eliminate the conflicting course sections. A section is selected randomly from the field of possible choices. The next course request is considered to determine the time choices that do not conflict with the previously assigned sections. The time choice can be random or on the basis of a weight applied according to the proportion of remaining seats in each possible section.

Several workable schedules are produced after a set number of trials, and the best schedule is selected according to a measure of relative goodness of each one.

Conflicts arise when the student's conflict matrix indicates no schedule is possible, and when no possible schedule is found after a certain number of attempts to fit the possible course sections into a workable schedule.

Pack's procedure is to select a teacher and room combination for a particular course, and then to find the students who are free at the same time period chosen. Pack's scheduling model includes the ranking of courses by enrollment, largest to smallest. The first step is to find the available times of the first choice teacher for a particular course. If the teacher does not have enough free time periods to satisfy the course section requirements, the second choice teacher's availability vector is scanned. Once a teacher is found, a room is found with the same free time. If a specific room had been designated for that course, its availability vector is the only one checked.

Now a search of the student course vectors locates the students requesting the course. Their free time vector indicates the number of available students. If this number does not exceed the room capacity, the teacher, room, and students are assigned to the selected time periods. This procedure is performed on each course in turn.

Teacher-room conflicts occur when no teacher-room combination has enough available time to satisfy a section's time requirement.

Tentative student conflicts may be resolved by an interchange of students between sections. If it is a single section course and all students requesting the course are not free, the free students are scheduled, with output messages indicating the section and the unscheduled students. It may be possible to resolve a conflict when a course has multi-sections. The conflict resolution techniques involve boolean operations of binary vectors.

A specific output of Pack's scheduling model is a tabulation of the time periods, room, and teacher for each section. Various outputs are in vector formats which describe the student-course requests, student free time, and teacher-room availability.

The output of Holz's GASP program gives the completed master schedule, with the number of students assigned to the course section and the remaining seats. A summary report on the resource utilization can also be included. Listings pertaining to course sections and time patterns for staff are printed out. Student schedules are on tape and must have computer programs written to obtain them as outputs.

While these two master schedule generating programs do not guarantee an optimal schedule (one which minimizes the number of conflicts), nevertheless, they have proven in their field tests to be of considerable value to school administrators in the construction of more efficient master schedules.

MATHEMATICAL APPROACH

The heuristic programs discussed in the previous sections actually simulate to a certain extent the manual scheduling procedures. Direct mathematical approaches have been attempted to develop master schedules; however, due to computer limitations on the solution of large scale scheduling problems, the methods have not been utilized to generate master schedules for real life problems.

Gotlieb has constructed small time-tables by iteration involving boolean matrices. In the discussion it is assumed that the weekly requirements could be sub-divided into five sets of daily requirements. This assumption sets the pattern structure for each day of the week. Also to ensure that certain critical facilities (e.g. lunch rooms, science classes, and gymnasia) are fully utilized, some part of the time-table is filled out in advance. Thus, Gotlieb's construction is one of completing a partially filled time table. In his method, the problem is first checked for feasibility with respect to the initial conditions. The tight combinations of teachers, classes, and hours are resolved systematically. At the end a set of final availability vectors which produce no new tight combinations are obtained. The argument shows that the existence of the final availability vectors is necessary for a feasible solution. The sufficiency of the final availability vectors is conjectured, but it is not shown. If the conjectured sufficiency is correct, then the decomposition which is made into tight combinations will be a feasible set of matrices. Thus, a schedule is readily constructed from the final availabilities. It is stated that the algorithm described has been tested by hand calculations and by limited computer runs on small time-tables and it yields valid results.

Sherman has formulated the combinatorial problem arising from scheduling university classes. His attempt is to find a partition of a finite set which maximizes a set function. The criterion adopted for selecting a many to one set function is the minimization of the expected number of conflicts for a randomly drawn student. The problem of selecting the best set function amounts to finding an optimal set partition of a finite set. A stochastic type algorithm is developed which can be used to obtain optimal or near optimal partitions. It is stated that the model presented gives rise to a combinatorial problem which, with the use of the best known methods of solution, is intractable when optimality is insisted.

4. PROCEDURE, ANALYSES OF DATA, AND FINDINGS

PROCEDURE, ANALYSES OF DATA, AND FINDINGS

ALGORITHMIC SCHEDULER

BOOLEAN ISOMORPHIC STRUCTURES

BY

A.G. HOLZMAN
I.B. TURKSEN

MODEL

The class scheduling system is comprehended in the organic structure and educational philosophy of the total educational institution. Naturally, varying environments give birth to varying structures and philosophies. While it would be impractical to consider a model which embraces all possible environments, nevertheless, certain basic components influence all educational scheduling systems. The general model which depicts these characteristic modules of a scheduling system is shown in Figure 1. The model may be segregated into two entities, inputs and outputs.

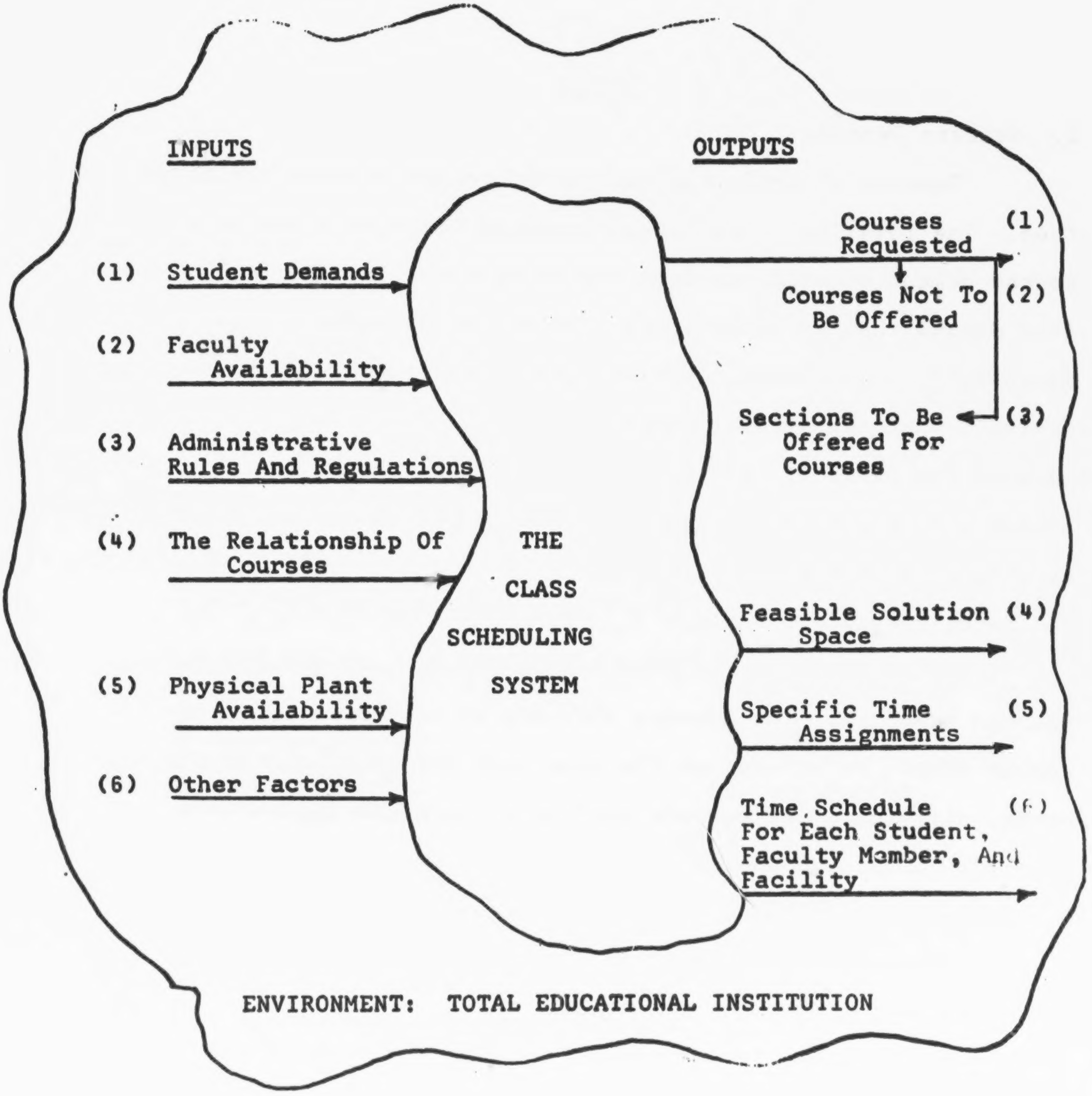

INPUTS
OUTPUTS
(1) Student Demands
(2) Faculty Availability
(3) Administrative Rules And Regulations
(4) The Relationship Of Courses
(5) Physical Plant Availability
(6) Other Factors
THE CLASS SCHEDULING SYSTEM
Courses Requested (1)
Courses Not To Be Offered (2)
Sections To Be Offered For Courses (3)
Feasible Solution Space (4)
Specific Time Assignments (5)
Time Schedule For Each Student, Faculty Member, And Facility (6)
ENVIRONMENT: TOTAL EDUCATIONAL INSTITUTION

Figure 1

Inputs

1. Student Demands

Demands of individual students are the courses requested by them. The totality of all these requests constitute the total course demand of students from the educational institution. It is this demand that an educational institution attempts to satisfy in the class scheduling system. Current practices indicate that administrators estimate what the student demand for each course will be for an incoming term on the basis of past experience and the present trend. It is quite possible to start initially with student demand which can be obtained by an earlier preregistration.

2. Faculty Availability

No course can be offered unless there is a faculty member who can teach it. The courses that are to be taught by a faculty member cannot be offered at the same time period of a day. In other words, different time periods must be assigned to the different courses taught by a faculty member.

3. Administrative Rules and Regulations

These are the parameters which designate the nature of administrative policies. For example,

a. Minimum Requirement Parameter. A minimum number of student requests must be obtained before a course is offered. This requirement establishes the minimum requirement parameter (MINR). If the number of students who request a course is less than MINR, that course is not offered. If, however, it is equal to or greater than MINR, then the course is offered.

b. Maximum Requirement Parameter. A maximum number of students is stipulated for each course. Once the courses to be given are determined, it is necessary to determine how many sections of each must be offered. The number of sections of a course can be found if the maximum requirement parameter (MAXR) and the total number of requests for this course are known.

4. The Relationship Of Courses

The set of courses requested by a student must not conflict, and therefore these courses must be scheduled at different time periods.

Courses which are co-requisites should not be offered at the same time period of a day, since a student who takes a co-requisite pair must be able to schedule these courses at different time periods.

Service or inter-departmental courses should be offered during time periods which facilitate the non-conflicting schedule establishment for students of related departments.

For these reasons, it is desirable to know the relations between the courses so that the non-conflicting time periods can be determined. This relationship of courses is reflected by the student demands.

5. Physical Plant Availability

The number of lecture halls, classrooms, and laboratories available at a given hour establishes limits on the maximum number of course sections that can be offered at a given time.

6. Other Factors

Other input factors to a class scheduling system also exist. For instance, the student course demand is taken to be an input. The origin of this demand can be based on the historical status of the students which could very well be developed into a tape oriented search-sort system. This status system could be made a part of the class scheduling system. Another factor could be a special structure of the schedule, such as fixed or flexible. A fixed schedule is one in which the same course is assigned the same time period every time it meets, whereas, a flexible schedule is one in which the same course may be assigned different time periods. Thus, it is conceivable that other factors can enter the environment.

Outputs

The outputs of the scheduling system are the results to be obtained. Typical outputs from the Class Scheduling System are shown in Figure 1.

1. Courses Requested

It is assumed here that the student demand is known. However, it is possible that this information may be obtained from a preregistration procedure, or from a master tape which contains a priority listing of the remaining courses in a student's program of study.

Given the individual student demands, a computer program was designed to determine a complete listing of all the courses to satisfy the total student demand. The computer also prepares a frequency count for each of the courses requested.

2. Courses Not To Be Offered

As stated in the discussion on inputs to the system, administrative rules usually specify the minimum requirement parameter (MINR) and the maximum requirement parameter (MAXR) for each of the courses. The demand counter for a particular course is compared with its MINR. If the demand is less than MINR, then the course number and the demand count is transferred to a list called Courses Not To Be Offered. It is also possible that a course can not be offered if a faculty member is not available to teach the course.

3. Sections For Courses To Be Offered

If the demand for a course is equal to or greater than its MINR, then the course is to be offered and the number of sections for this course is determined by $\frac{\text{Demand}}{\text{MAXR}}$. It should be pointed out that some of the outputs have feed back effects on the system. For example, if a student has requested a course that is not to be given, then he must request another course which is to be offered.

4. Feasible Solution Space

The feasible solution space indicates sets of courses which can be scheduled at the same time period without violating the constraints imposed on the scheduling system by the student demands, faculty availability, and physical plant availability. As stated earlier, the prime objective of this research is to determine the feasible solution area. Consequently, this will be discussed in considerable detail later.

5. Specific Time Assignments

Once the feasible solution space is determined, the next step is to assign specific time periods to the set of courses which do not violate the restrictions of the system. For example, one subset of courses may be assigned to the first period, another subset to the second period, etc.

6. Time Schedule For Each Student, Faculty Member And Facility

If the constraints on the system comprehend the total student demand, faculty availability, and physical plant facility, then the feasible solution space will guarantee a time schedule free of conflicts for each student. Since specific time period assignments can be made to non-conflicting sets of courses, as stated in (5) then time schedules for each student, faculty member and facility can be established.

BEHAVIOR OF THE BOOLEAN STRUCTURE

It has been shown that the courses to be offered and the number of sections of each can be determined easily. At this stage, however, the sections of all courses have no relation and order among themselves. The following principles of well ordering have been developed so that the ordering is amenable to the algorithms which are developed later.

1. Principles Of Well Ordering

Each course section is associated with a symbol A_s, where $s = 1, 2, \ldots, \mu$ = total number of sections of all the courses an educational institution is to offer for a specific term.

Notation $CA_{(i),k}$ is used to indicate the set of k sections of a given course $CA_{(i)}$, where $i = 1, 2, \ldots, w$ = total number of courses, $k = 1, 2, \ldots, k_i$ = total number of sections for the i th course.

a. Principle 1. The sections of a course are numbered by consecutive positive integers. If a course is $CA_{(i)k}$, then its sections will be symbolized by $A_{j+1}, A_{j+2}, \ldots, A_{j+k-1}, A_{j+k}$, where $j \geq 0$, $j+k \leq \mu$, and j is an arbitrary starting point which will be specified later.

b. Principle 2. If a set of courses is $CA_{(i),p}$, $CA_{(i+1),q}$, $CA_{(i+2),r}$, each having p, q, r, sections with $p \geq q \geq r$, and there exists no $CA_{(i+x),v}$, such that $q<v<p$ or $r<v<q$, then the courses can be ordered as follows;

$CA_{(i),p}$: $A_{j+1}, A_{j+2}, \ldots, A_{j+p}$
$CA_{(i+1),q}$: $A_{j+p+1}, A_{j+p+2}, \ldots, A_{j+p+q}$
$CA_{(i+2),r}$: $A_{j+p+q+1}, A_{j+p+q+2}, \ldots, A_{j+p+q+r}$

Example: Suppose a set $S = [CA_{(i),6}, CA_{(i+1),5}, CA_{(i+2),3}, CA_{(i+3),1}, CA_{(i+4),1}]$, then for a starting point of arbitrary j;

$CA_{(i),6}$: $A_{j+1}, A_{j+2}, A_{j+3}, A_{j+4}, A_{j+5}, A_{j+6}$
$CA_{(i+1),5}$: $A_{j+7}, A_{j+8}, A_{j+9}, A_{j+10}, A_{j+11}$
$CA_{(i+2),3}$: $A_{j+12}, A_{j+13}, A_{j+14}$
$CA_{(i+3),1}$: A_{j+15}
$CA_{(i+4),1}$: A_{j+16}

c. Principle 3. The course which has the maximum number of sections among all the courses will have a section symbolized by A_1.

Example: Suppose the set S is as defined in the example for Principle 2. The $CA_{(i),6}$ has the maximum number of sections among the courses of this set, and thus by Principle 3 it will have a section called A_1. By Principle 1, the remaining sections are numbered consecutively. (It is obvious that $j=0$.)

$CA_{(1),6}$: $A_1, A_2, A_3, A_4, A_5, A_6$

Continuing by Principle 2, we obtain the following ordering;

$CA_{(2),5}$: $A_7, A_8, A_9, A_{10}, A_{11}$

$CA_{(3),3}$: A_{12}, A_{13}, A_{14}

$CA_{(4),1}$: A_{15}

$CA_{(5),1}$: A_{16}

If there is more than one course which has the maximum number of sections, then one of them is arbitrarily selected to have a section symbolized by A_1.

2. Scheduling Variables And Functions

a. Variables. The variables of the class scheduling system are the well ordered sections of courses defined previously. Each course section A_s is a two-valued variable with respect to time assignment. If A_s is offered at time period t, then $A_s = 1$. If A_s is not offered at time period t, then $A_s = 0$. With μ two-valued variables $A_1, A_2, \ldots, A_\mu$, there are 2^μ configurations. These configurations are formed by permitting every variable to take on each of its two values (0,1) in all possible combinations. Each configuration is represented by a sequence of μ binary digits, and hence correspondence to a positive integer i,[1] $0 \leq i \leq 2^\mu - 1$. Such a representation will be called a vector.

---

[1] The integral domain of integers is usually considered as consisting of positive integers, zero, and negative integers. However, by convention, we will consider zero as a positive integer in this paper.

b. Functions. A class scheduling function f_j is a two-valued function of two or more of the variables A_s defined above. The function $f_j = 0$ is called the zero function; and $f_j = 1$ is called the unit function.

The zero functions in particular will be called the constraints of the class scheduling problem. As indicated earlier, there are 2^μ configurations associated with μ variables. These 2^μ configurations make up the total solution space. The zero functions enable us to obtain the desirable configurations of space. Three distinct classes of constraints are constituted by the zero functions.

3. Classes Of Functions And Related Theorems

Class I. Assume: A_α is the only section of $CA_{(i+x),1}$ and A_β is the only section of $CA_{(i),1}$ with $(i+x) > (i)$, which means $\alpha > \beta$ by the well ordering principle stated previously.

$f\,(CA_{(i+x),1}, CA_{(i),1}) = A_\alpha \cdot A_\beta = 0$ shall be interpreted as: The course section A_α AND the course section A_β cannot be scheduled at the same time period. The operation $(\cdot)$ is associated with Boolean logical intersection which is called AND.

Class II. Assume: $A_\beta, A_{\beta-1}, \ldots, A_{\beta-r+1}$ are the r sections of $CA_{(i),r}$ and A_α is the only section of $CA_{(i+x),1}$. Thus $(i+x) > (i)$, and $\alpha > \beta$ by the well ordering principles.

$f\,(CA_{(i+x),1}, CA_{(i),r}) = A_\alpha \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-r+1}) = 0$ shall be interpreted as: The course section A_α AND the r sections $A_\beta, A_{\beta-1}, \ldots, A_{\beta-r+1}$, can not be scheduled at the same time period. The implication here is such that if A_α is offered at time period t, one or all sections of $CA_{(i),r}$ can be offered at any time period except the t th one.

The operation (+) is associated with Boolean logical union which is called "OR".

Class III. Assume: $A_\alpha, A_{\alpha-1}, \ldots, A_{\alpha-k+1}$ are the k sections of $CA_{(i+x),k}$ and $A_\beta, A_{\beta-1}, \ldots, A_{\beta-r+1}$ are the r sections of $CA_{(i),r}$. Thus $(i+x)>i$, and $(\alpha-k+1)>\beta$, by the well ordering principles.

$f(CA_{(i+x),k}, CA_{(i),r}) = (A_\alpha + A_{\alpha-1} + \cdots + A_{\alpha-k+1}) \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0$ shall be interpreted as: The sections of the course $CA_{(i+x),k}$, AND the sections of the course $CA_{(i),r}$ cannot be scheduled at the same time period.

Theorem I. All Class III constraints can be reduced to Class II constraints.

$$(A_\alpha + A_{\alpha-1} + \cdots + A_{\alpha-k+1}) \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0$$

can be expanded by the distributive law of Boolean algebra as follows: $A_\alpha \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) + A_{\alpha-1} \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) + \cdots + A_{\alpha-k+1} \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0$. For an "OR" function to be "0", each of the multiplicative terms must be equal to zero. Thus,

$$\begin{array}{ll}
A_\alpha \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) & = 0 \\
A_{\alpha-1} \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) & = 0 \\
\cdot \quad \cdot \quad \cdots \quad \cdot & \cdot \\
\cdot \quad \cdot \quad \cdots \quad \cdot & \cdot \\
\cdot \quad \cdot \quad \cdots \quad \cdot & \cdot \\
A_{\alpha-k+1} \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) & = 0
\end{array}$$

This final set of equations belongs to Class II. q.e.d.

Theorem II. All Class II constraints can be reduced to Class I constraints.

$A_\alpha \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0$ belongs to Class II.

By the distributive law of Boolean algebra, this expression can be stated as follows:

$$A_\alpha \cdot A_\beta + A_\alpha \cdot A_{\beta-1} + \cdots + A_\alpha \cdot A_{\beta-r+1} = 0$$

Again for an "OR" function to be "0" each of the multiplicative terms must be equal to zero. Therefore,

$$\begin{array}{lll}
A_\alpha \cdot A_\beta & = & 0 \\
A_\alpha \cdot A_{\beta-1} & = & 0 \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
A_\alpha \cdot A_{\beta-r+1} & = & 0
\end{array}$$

This final set of equations belongs to Class I. q.e.d.

4. Correspondence Between The Model And The Zero Functions

Constraints in the scheduling model are determined from restrictions imposed by student demands, faculty availability, relationship of courses, and physical plant availability.

If there are n distinct course sections taught by a faculty member, then these constraints can be expressed as the zero functions that belong to Class I. Each pair of course sections make up a constraint. For n course sections there are $C_2^n = \frac{n!}{(n-2)!\,2!}$ zero functions.

Example: Suppose a faculty member is to teach three course sections A_j, A_q, A_k, $(j>q>k)$. The zero functions are:

$$A_j \cdot A_q = 0$$
$$A_j \cdot A_k = 0$$
$$A_q \cdot A_k = 0$$

Likewise, constraints established by student demands, relationship of courses, and faculty availability, are expressed at zero functions.

Examples:

(a) If the only sections of two courses which can not be offered at a given time period are A_α and A_β, with $\alpha > \beta$, then $A_\alpha \cdot A_\beta = 0$.

(b) If a single section course $CA_{(i+x),1} = A_\alpha$ and a multiple section course $CA_{(i),r} = (A_\beta, A_{\beta-1}, \ldots, A_{\beta-r+1})$, can not be offered at the same time period, then $A_\alpha \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-r+1}) = 0$, $\alpha > \beta$.

(c) If a multiple section course $CA_{(i+x),k} = A_\alpha, A_{\alpha-1}, \ldots, A_{\alpha-k+1}$, and another multiple section course $CA_{(i),r} = (A_\beta, A_{\beta-1}, \ldots, A_{\beta-r+1})$ can not be offered at the same time period, with $(\alpha-k+1) > \beta$, then we have the following set of constraints as a result of Theorem 1.

$$\begin{array}{ll}
A_\alpha & \cdot\ (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0 \\
A_{\alpha-1} & \cdot\ (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r-1}) = 0 \\
\cdot & \quad \cdot \quad \cdot \quad \cdots \quad \cdot \qquad \cdot \\
\cdot & \quad \cdot \quad \cdot \quad \cdots \quad \cdot \qquad \cdot \\
\cdot & \quad \cdot \quad \cdot \quad \cdots \quad \cdot \qquad \cdot \\
A_{\alpha-k+1} & \cdot\ (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0
\end{array}$$

A constraint such as $A_\alpha \cdot (A_\beta + A_{\beta-1} + \cdots + A_{\beta-r+1}) = 0$ may be too severe. This will occur if r-s sections (with $s < r$) of $CA_{(i),r}$ can be offered in the same time period as A_α.

Thus only s sections of $CA_{(i),r}$ and A_α must not conflict. When this is the case, $A_\alpha \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-r+1}) = 0$ reduces to $A_\alpha \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-s+1}) = 0$.

Example Of The Boolean Structure

Suppose there exists four courses $CA_{(1),2} = [A_2, A_1]$; $CA_{(2),1} = [A_3]$; $CA_{(3),1} = [A_4]$, and $CA_{(4),1} = [A_5]$. Assume the following constraints between the course sections exist.

$$\begin{aligned} A_5 &\cdot A_4 = 0 \\ A_5 &\cdot A_3 = 0 \\ A_4 &\cdot A_3 = 0 \\ A_3 &\cdot A_2 = 0 \\ A_5 &\cdot (A_2 + A_1) = 0 \\ A_4 &\cdot (A_2 + A_1) = 0 \end{aligned}$$

Since there are 5 variables in the problem, the number of column configurations = $2^5 = 32$, with each configuration corresponding to one of the positive integers 0, 1, 2,..., 31. Refer to Tableau 1. With every variable we associate a designation number. As an example, for variable A_1 the designation number is 0101 0101 ... 0101. The set of designation numbers assigned to all variables is called a basis. The designation number for A_1 is obtained simply by alternating one zero and one unit, for A_2 by alternating two zeros and two units, for A_3 by alternating four zeros and four units, etc. The 32 columns of Tableau 1 are the total possible configurations for $A_1, A_2, \ldots, A_5$, ignoring constraints. Each column of this basis, read from bottom to top, forms a binary number which corresponds to the positive integer at the top of the column.

Positive Integer Correspondence for the Boolean Configurations

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Variables	A_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	A_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	A_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	A_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	A_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tableau 1

Functional Values of the Constraints

Functions	$f_1=A_5\cdot A_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	$f_2=A_5\cdot A_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	$f_3=A_4\cdot A_3$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	$f_4=A_3\cdot A_2$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
	$f_5=A_5\cdot(A_2+A_1)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
	$f_6=A_4\cdot(A_2+A_1)$	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
$f_1+f_2+f_3+f_4+f_5+f_6$		0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tableau 2

The Boolean operations are defined as follows:

	A	B	A•B
AND	0	0	0
	0	1	0
	1	0	0
	1	1	1

	A	B	A+B
OR	0	0	0
	0	1	1
	1	0	1
	1	1	1

Referring to Tableau 2, when $f_1 + f_2 + f_3 + f_4 + f_5 + f_6 = 0$, all the necessary constraints of the problem are satisfied. Hence, the feasible solution space is denoted by the zeros in the final row of Tableau 2. The columns in the basis of Tableau 1 which correspond to the zeros in the final row of Tableau 2 represent the feasible solution space for this scheduling problem. The following Tableau 3, which was abstracted from Tableau 1, is the feasible solution space.

Positive Integer Correspondence

		0	1	2	3	4	5	8	16
	A_1	0	1	0	1	0	1	0	0
	A_2	0	0	1	1	0	0	0	0
Variables	A_3	0	0	0	0	1	1	0	0
	A_4	0	0	0	0	0	0	1	0
	A_5	0	0	0	0	0	0	0	1

Feasible Solution Space

Tableau 3

The units in a column configuration of the feasible solution space designate the courses which can be offered at the same time period without violating any of the restrictions on the system.

Time assignments can be made by the following algorithm.

(1) Start with a column, and assign time period t_1 to all the variables which have a 1 in this column.

(2) Mark off the rows of the variable (s) which had a time assignment in this column.

(3) Select a different column and repeat (1) and (2). Repeat this procedure until all of the variables have a time assignment.

For the above problem: (Refer to Tableau 3)

Column 16: Assign A_5 to time period t_1. Mark off row A_5.

Column 8: Assign A_4 to time period t_2. Mark off row A_4.

Column 5: Assign A_1 and A_3 to time period t_3. Mark off the rows A_3 and A_1.

Column 4: No new assignment since A_3 has all ready been assigned.

Column 3: Assign A_2 to time period t_4. Mark off row A_2.

All the rows now have been marked off; hence the assignment of courses to time periods ends.

This manual procedure becomes exceedingly cumbersome and practically impossible as the number of variables μ becomes large. However, algorithms have been developed and programmed for computer solution to make the necessary computations.

THE ALGORITHMS

The one to one correspondence between the column configurations of the Boolean basis and the positive integers is shown in Tableau 1 of the previous example. A rigorous algebraic proof can be given and the isomorphism between the columns of the Boolean basis and the positive integers can be shown. It will be

It is possible to continue the example for each of the restrictions and to show this new method of calculation for all the restrictions. After a period of experimentation, the cyclic behavior of the subsets which are equivalent to the feasible solution space of a constraint can be discovered. The experimentations made possible the construction of empirical algorithms. A combinational Fortran-Autocoder program for IBM 7070 was designed to determine the cyclic behavior of the subsets of the feasible solution space.

Mathematical Algorithms For The Calculation Of The Subsets For Class I Constraints

Suppose a restriction is $A_\alpha \cdot A_\beta = 0$, with $\alpha > \beta$, in a problem with the total number of course sections $= \mu$. The subsets which satisfy the constraints are denoted by double subscript notation. The first subscript denotes the cycle and the second subscript denotes the subset number of that particular cycle.

$$\text{Subset}_{1,1} = [\overbrace{0}^{B_1},\ \overbrace{2^{\alpha-1}+\underset{h}{2^{\beta-1}}-1}^{E_1}]$$

$$\text{Subset}_{1,2} = [\overbrace{2^{\alpha-1}+1\cdot 2^{\beta}}^{B_2},\ \overbrace{B_2+h}^{E_2}]$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\text{Subset}_{1,j} = [\overbrace{2^{\alpha-1}+(j-1)\cdot 2^{\beta}}^{B_j},\ \overbrace{B_j+h}^{E_j}] \qquad\qquad \text{1st cycle}$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\text{Subset}_{1,n} = [\overbrace{2^{\alpha-1}+(n-1)\cdot 2^{\beta}}^{B_n},\ \overbrace{B_n+h}^{E_n}]$$

Number of subsets in cycle $= n = 2^{\alpha-1-\beta}$

sufficient, however, to indicate that there is one and only one positive integer for each Boolean column configuration specified here. Furthermore, this correspondence is unique, single valued, onto, and operation preserving.

For the construction of Tableau 2, it was necessary to operate on each element of the rows of the variables specified by a constraint. The operations involved are Boolean AND and OR. The cyclic nature of the Boolean configurations suggests the possibility of another procedure for the determination of the zeros of a given constraint. Take for instance, $f_1 = A_5 \cdot A_4 = 0$. We see from Tableau 2, that $f_1 = 0$ within the subset $\{0,1,2,...,23\}$ and nowhere else. For simplicity, a subset such as $\{0,1,2,...,23\}$ will be denoted by $[0,23]$, which indicates the initial element and the terminal element of the subset of all integers from 0 through 23. The subset $[0, 23]$ can be realized also in the following way, $[0, 2^{5-1} + 2^{4-1} -1] = [0, 16 + 8-1] = [0, 23]$.

The zeros for $f_4 = A_3 \cdot A_2 = 0$ are the subsets $[0, 5]$, $[8, 13]$, $[16, 21]$, $[24, 29]$. The subset $[0, 5] = [0, 2^{3-1} + 2^{2-1} -1] = [0, 4 + 2 - 1] = [0, 5]$. For the subset $[8, 13]$, one could conjecture that $[1 \cdot 2^3 + \underline{0}, 1 \cdot 2^3 + \underline{5}] = [8, 13]$, where $\underline{0}$ and $\underline{5}$ are the respective values of the beginning and end points of the subset $[0, 5]$.

For subset $[16, 21]$, one could conjecture that $[2 \cdot 2^3 + \underline{0}, 2 \cdot 2^3 + \underline{5}] = [16, 21]$, where $\underline{0}$ and $\underline{5}$ are again the initial and terminal points of the subset $[0, 5]$.

For the subset $[24, 29]$, one could conjecture that $[3 \cdot 2^3 + 0, 3 \cdot 2^3 + 5] = [24, 29]$ which is again the same as given by the tableau.

$\text{Subset}_{2,1} = [1\cdot 2^{\alpha}+B_1,\ 1\cdot 2^{\alpha}+E_1]$

$\text{Subset}_{2,2} = [1\cdot 2^{\alpha}+B_2,\ 1\cdot 2^{\alpha}+E_2]$

$\vdots \qquad \vdots \qquad \vdots$

$\text{Subset}_{2,j} = [1\cdot 2^{\alpha}+B_j,\ 1\cdot 2^{\alpha}+E_j]$ 2nd cycle

$\vdots \qquad \vdots \qquad \vdots$

$\text{Subset}_{2,n} = [1\cdot 2^{\alpha}+B_n,\ 1\cdot 2^{\alpha}+E_n]$

Number of subsets in cycle = $n = 2^{\alpha-1-\beta}$

$\text{Subset}_{3,1} = [2\cdot 2^{\alpha}+B_1,\ 2\cdot 2^{\alpha}+E_1]$

$\text{Subset}_{3,2} = [2\cdot 2^{\alpha}+B_2,\ 2\cdot 2^{\alpha}+E_2]$

$\vdots \qquad \vdots \qquad \vdots$

$\text{Subset}_{3,j} = [2\cdot 2^{\alpha}+B_j,\ 2\cdot 2^{\alpha}+E_j]$ 3rd cycle

$\vdots \qquad \vdots \qquad \vdots$

$\text{Subset}_{3,n} = [2\cdot 2^{\alpha}+B_n,\ 2\cdot 2^{\alpha}+E_n]$

Number of subsets in cycle = $n = 2^{\alpha-1-\beta}$

For the i th cycle, then

$$\text{Subset}_{i,1} = [(i-1)\cdot 2^{\alpha}+B_1,\ (i-1)\cdot 2^{\alpha}+E_1]$$

$$\text{Subset}_{i,2} = [(i-1)\cdot 2^{\alpha}+B_2,\ (i-1)\cdot 2^{\alpha}+E_2]$$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \end{array}$$

$$\text{Subset}_{i,j} = [(i-1)\cdot 2^{\alpha}+B_j,\ (i-1)\cdot 2^{\alpha}+E_j] \qquad \text{i th cycle}$$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \end{array}$$

$$\text{Subset}_{i,n} = [(i-1)\cdot 2^{\alpha}+B_n,\ (i-1)\cdot 2 +E_n]$$

Number of subsets in cycle = $n = 2^{\alpha-1-\beta}$

For the last cycle, the m th cycle, then

$$\text{Subset}_{m,1} = [(m-1)\cdot 2^{\alpha}+B_1,\ (m-1)\cdot 2^{\alpha}+E_1]$$

$$\text{Subset}_{m,2} = [(m-1)\cdot 2^{\alpha}+B_2,\ (m-1)\cdot 2^{\alpha}+E_2]$$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \end{array}$$

$$\text{Subset}_{m,j} = [(m-1)\cdot 2^{\alpha}+B_j,\ (m-1)\cdot 2^{\alpha}+E_j] \qquad \text{m th cycle}$$

$$\begin{array}{ccc} \vdots & \vdots & \vdots \end{array}$$

$$\text{Subset}_{m,n} = [(m-1)\cdot 2^{\alpha}+B_n,\ (m-1)\cdot 2^{\alpha}+E_n]$$

Number of subsets in cycle = $n = 2^{\alpha-1-\beta}$

Since μ = total number of course sections = total number of variables in the problem, then, for a given constraint, the total number of cycles $m = 2^{\mu-\alpha}$.

Since by Theorems I and II all Class III constraints are reducible to Class II constraints, and all Class II constraints are reducible to Class I constraints, the above algorithms can be used for any class constraint.

On The Class II Constraint

While a Class II restriction can be reduced to a Class I restriction (Theorem II), nevertheless, with one modification in the algorithms developed for Class I constraints, it is possible to operate directly on the Class II constraint,

$$A_\alpha \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-r+1}) = 0,$$

with $\alpha > \beta$, and the subscripts β, $\beta-1, \ldots, \beta-r+1$, numbered consecutively with $\beta > \beta-1 > \ldots > \beta-r+1$. Leaving $\eta = \beta-r+1$, the only change in the algorithms is the calculation for

$$\text{Subset}_{1,1} = [0,\ 2^{\alpha-1} + 2^{\eta-1} - 1],$$

where η replaces β in the calculation. The remaining subsets are obtained by the algorithms used for a Class I constraint.

The number of subsets in a cycle equals $2^{\alpha-1-\beta}$, and the number of cycles equals $2^{\mu-\alpha}$, the same as stated previously for a Class I constraint.

On The Class III Constraint

While a Class III restriction can be reduced to a Class II restriction (Theorem I), and a Class II restriction can be reduced to a Class I restriction (Theorem II), the algorithms developed for Class I constraints can be modified to operate directly on the Class III constraint,

$$(A_\alpha + A_{\alpha-1} + \ldots + A_{\alpha-k+1}) \cdot (A_\beta + A_{\beta-1} + \ldots + A_{\beta-r+1}) = 0, \text{ with}$$

$(\alpha-k+1) > \beta$. The subscripts α, $\alpha-1, \ldots, \alpha-k+1$, are numbered consecutively with $\alpha > \alpha-1 > \ldots > \alpha-k+1$; the subscripts β, $\beta-1, \ldots, \beta-r+1$ are numbered consecutively with $\beta>\beta-1>\ldots>\beta-r+1$. Leaving $\phi=\alpha-k+1$, and $\eta=\beta-r+1$, the algorithm for the calculation of

$$\text{Subset}_{1,1} = [0,\ 2^{\phi-1}+2^{\eta-1}-1],$$

where ϕ replaces α and η replaces β as used for the Class I constraint. The algorithm for the calculation of the j th subset ($j\neq 1$) of the first cycle,

$$\text{Subset}_{1,j} = [2^{\phi-1}+(j-1)\cdot 2^{\beta},\ B_j+h]$$

where ϕ replaces α as used for the Class I constraint. The algorithms to determine the subsets for cycles 2, 3,..., m, are the same as developed for the Class I constraint.

The number of cycles equals $2^{\mu-\alpha}$, the same as for Class I and Class II constraints; however, the number of subsets in a cycle $n=2^{\alpha-1-\beta}+2^{(\alpha-1)-1-\beta}+2^{(\alpha-2)-1-\beta}+\ldots+2^{\phi-1-\beta}$, instead of $2^{\alpha-1-\beta}$ as used to determine the algorithms for the Class I and Class II constraints.

For example, consider the constraint, $(A_7+A_6+A_5)\cdot(A_3+A_2+A_1) = 0$, where the total number of variables in the problem is 7. Then $\mu=7$,

$\alpha=7$, $\phi=5$

$\beta=3$, $\eta=1$

The number of cycles $m = 2^{7-7} = 1$, and the number of subsets in the cycle $n = 2^{7-1-3}+2^{6-1-3}+2^{5-1-3} = 14$.

The Hyper Cube Of The Feasible Solution Of The μ Variable Problem

General Algorithm for the Computer Program

Step 1. Process the following for each constraint.

a. Determine the subsets for the constraint by the previously developed mathematical algorithms in the domain of positive integer algebra.

Determine n, the number of subsets in a cycle for this constraint.

Determine m, the number of cycles for this constraint.

b. Restrict the solution subsets constructed from the previous constraints by the subsets of this constraint.

Feasible Solution Subsets =
(including this constraint)

Feasible Solution Subsets $\cap$ Subsets Of This Constraint
(previous to this constraint)

Repeat Step 1 until feasible solution subsets include all constraints of the problem. When Step 1 processes all the given constraints, the result will be a number of subsets in positive integer domain where the solution lies.

Step 2. Map the members of subsets which are in positive integer domain to the domain of positive binary numbers. The correspondence is isomorphic. That is to say there will be a binary configuration for each member of the subsets, and this configuration will be unique. These binary row configurations determine the vertices of a μ-dimensional hyper cube where the feasible solution lies.

Hypothetical Example

Assume there are ten courses to be scheduled in three blocks, with no conflicts permitted between time assignments of the courses within the block. The relationship of the courses, derived from the block structures, establishes the constraints of the problem.

Faculty constraints and physical plant constraints could also be included here, but are ignored in this hypothetical case to avoid disguising the methodology in a multitude of constraints. Constraints imposed by faculty availability and physical plant availability would be added to those shown here.

Set of Courses to be Scheduled	Block 1	Block 2	Block 3
A_1 A_2	A_1 A_2	A_3 A_4	A_5 A_6
A_3 A_4	A_7	A_7	A_7
A_5 A_6	A_8	A_9	A_9
A_7	A_9	A_{10}	
A_8			
A_9			
A_{10}			

NOTE: A_1 and A_2, A_3 and A_4, A_5 and A_6, are sections of the same course.

Restrictions for Block 1

$A_9 \cdot A_8 = 0$

$A_9 \cdot A_7 = 0$

$A_8 \cdot A_7 = 0$

$A_9 \cdot (A_2 + A_1) = 0$

$A_8 \cdot (A_2 + A_1) = 0$

$A_7 \cdot (A_2 + A_1) = 0$

Restrictions for Block 2

$A_{10} \cdot A_9 = 0$

$A_{10} \cdot A_7 = 0$

$A_9 \cdot A_7 = 0$

$A_{10} \cdot (A_4 + A_3) = 0$

$A_9 \cdot (A_4 + A_3 = 0$

$A_7 \cdot (A_4 + A_3 = 0$

Restrictions for Block 3

$A_9 . A_7 = 0$

$A_9 \cdot (A_6 + A_5) = 0$

$A_7 \cdot (A_6 + A_5) = 0$

For a 10 variable problem, there are $2^{10} = 1024$ possible configurations. However, due to the restrictions imposed on this problem, there are only 102 feasible configurations. A computer program in combined Fortran-Autocoder language was written to calculate the subsets by the algorithms previously developed. The 102 feasible configurations are contained in the following 26 subsets, with the isomorphic mapping from the positive integer number to the binary number also shown.

Integers	Binary									
	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
$S_1 = [0, 64]$										
0										0
1										1
2									1	0
3									1	1
4								1	0	0
5								1	0	1
6								1	1	0
7								1	1	1
8							1	0	0	0
9							1	0	0	1
10							1	0	1	0
11							1	0	1	1
12							1	1	0	0
13							1	1	0	1
14							1	1	1	0
15							1	1	1	1
16						1	0	0	0	0
17						1	0	0	0	1
18						1	0	0	1	0
19						1	0	0	1	1
20						1	0	1	0	0
21						1	0	1	0	1
22						1	0	1	1	0
23						1	0	1	1	1
24						1	1	0	0	0
25						1	1	0	0	1
26						1	1	0	1	0
27						1	1	0	1	1
28						1	1	1	0	0
29						1	1	1	0	1
30						1	1	1	1	0
31						1	1	1	1	1
32					1	0	0	0	0	0
33					1	0	0	0	0	1
34					1	0	0	0	1	0
35					1	0	0	0	1	1
36					1	0	0	1	0	0
37					1	0	0	1	0	1
38					1	0	0	1	1	0
39					1	0	0	1	1	1
40					1	0	1	0	0	0

Integers		Binary
41		1 0 1 0 0 1
42		1 0 1 0 1 0
43		1 0 1 0 1 1
44		1 0 1 1 0 0
45		1 0 1 1 0 1
46		1 0 1 1 1 0
47		1 0 1 1 1 1
48		1 1 0 0 0 0
49		1 1 0 0 0 1
50		1 1 0 0 1 0
51		1 1 0 0 1 1
52		1 1 0 1 0 0
53		1 1 0 1 0 1
54		1 1 0 1 1 0
55		1 1 0 1 1 1
56		1 1 1 0 0 0
57		1 1 1 0 0 1
58		1 1 1 0 1 0
59		1 1 1 0 1 1
60		1 1 1 1 0 0
61		1 1 1 1 0 1
62		1 1 1 1 1 0
63		1 1 1 1 1 1
64		1 0 0 0 0 0 0
	$S_2 = [128,128]$	
128		1 0 0 0 0 0 0 0
	$S_3 = [132,132]$	
132		1 0 0 0 0 1 0 0
	$S_4 = [136,136]$	
136		1 0 0 0 1 0 0 0
	$S_5 = [140,140]$	
140		1 0 0 0 1 1 0 0
	$S_6 = [144,144]$	
144		1 0 0 1 0 0 0 0
	$S_7 = [148,148]$	
148		1 0 0 1 0 1 0 0
	$S_8 = [152,152]$	
152		1 0 0 1 1 0 0 0
	$S_9 = [156,156]$	
156		1 0 0 1 1 1 0 0

Integers		Binary
	$S_{10} = [160,160]$	
160		1 0 1 0 0 0 0 0
	$S_{11} = [164,164]$	
164		1 0 1 0 0 1 0 0
	$S_{12} = [168,168]$	
168		1 0 1 0 1 0 0 0
	$S_{13} = [172,172]$	
172		1 0 1 0 1 1 0 0
	$S_{14} = [176,176]$	
176		1 0 1 1 0 0 0 0
	$S_{15} = [180,180]$	
180		1 0 1 1 0 1 0 0
	$S_{16} = [184,184]$	
184		1 0 1 1 1 0 0 0
	$S_{17} = [188,188]$	
188		1 0 1 1 1 1 0 0
	$S_{18} = [256,256]$	
256		1 0 0 0 0 0 0 0 0
	$S_{19} = [512,515]$	
512		1 0 0 0 0 0 0 0 0 0
513		1 0 0 0 0 0 0 0 0 1
514		1 0 0 0 0 0 0 0 1 0
515		1 0 0 0 0 0 0 0 1 1
	$S_{20} = [528,531]$	
528		1 0 0 0 0 1 0 0 0 0
529		1 0 0 0 0 1 0 0 0 1
530		1 0 0 0 0 1 0 0 1 0
531		1 0 0 0 0 1 0 0 1 1
	$S_{21} = [544,547]$	
544		1 0 0 0 1 0 0 0 0 0
545		1 0 0 0 1 0 0 0 0 1
546		1 0 0 0 1 0 0 0 1 0
547		1 0 0 0 1 0 0 0 1 1

Integers		Binary
	S_{22} = [560,563]	
560		1 0 0 0 1 1 0 0 0 0
561		1 0 0 0 1 1 0 0 0 1
562		1 0 0 0 1 1 0 0 1 0
563		1 0 0 0 1 1 0 0 1 1
	S_{23} = [640,640]	
640		1 0 1 0 0 0 0 0 0 0
	S_{24} = [656,656]	
656		1 0 1 0 0 1 0 0 0 0
	S_{25} = [672,672]	
672		1 0 1 0 1 0 0 0 0 0
	S_{26} = [688,688]	
688		1 0 1 0 1 1 0 0 0 0
	Boolean variables	A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1

The Boolean variables (i.e. the course sections) have the indicated correspondence with the positional values of the binary numbers.

The assignments are made by application of the algorithm on page 18.

Suppose the binary configuration corresponding to the integer 688 is selected. Using this configuration, course sections A_{10}, A_8, A_6, and A_5 are scheduled in the same time period. These course sections are then checked-off, indicating that they have already been scheduled.

The binary configuration corresponding to the integer 563 indicates that course sections A_{10}, A_6, A_5, A_2 and A_1 can be scheduled in the same period without conflicts. Since A_{10}, A_6, and A_5 had been scheduled previously, then only section A_2 and A_1 are scheduled in the same time period. A_2 and A_1 are then checked off.

In a similar manner, the configurations corresponding to integers 256, 188, and 64 indicate that Section A_9 is scheduled by itself in a time period, section A_4 and A_3 are scheduled together in a time period, and section A_7 is scheduled by itself.

The time assignments of the variables $A_{10}, A_9, \ldots, A_1$ which were made within the binary configurations of the positive integer elements, (688, 563, 256, 188, 64) of the previously obtained subsets, do not violate any of the constraints of this scheduling problem. The selection of the configurations was made arbitrarily by inspection; however, the problem of selection of binary configurations is very important and further research is needed in this area.

Since there are many possible combinations for time assignments of course sections, research should be conducted in this area to develop decision-making rules which will enable the best assignment to be made according to some criterion. The minimal number of time periods required to satisfy the constraints should also be investigated.

SUMMARY

The model of a general class scheduling system was given; its environment was explored, and a method of determining a master schedule was developed. A Boolean basis tableau approach was introduced: then, the cyclic behavior of the subsets of a given constraint and the isomorphic correspondence between the binary configurations and the positive integers led to the development of algorithms. The general algorithms programmed on the computer map the positive integer elements of the subsets onto the binary number column configuration. For the 10- variable problem, there are 2^{10} = 1024 such possible configuration. It was shown by the example that these 1024 configurations are reduced to 102 feasible configurations. Specific assignments were made by inspection of the feasible solution space.

A possible factor that may influence the starting point of the selection of configurations could be the number of classrooms or laboratories available. Another approach might be a probabilistic demand criterion which has some assigned probabilistic weight for each course section. Therefore, the assignment of the course sections with highest probability might take the priority.

It has been shown that by the use of Boolean isomorphic structures it is possible to determine the feasible solution space which does not violate any of the constraints of the problem. The solution was shown for a 10 variable problem; however, the program can be expanded to increase the number of variables so that an economic appraisal can be made on the complete time required for a realistic scheduling problem. Such an appraisal may suggest modifications in the approach presented to make it amenable to solution of an actual scheduling complex of students, faculty, and facilities.

RELATIONS MATRIX STRUCTURES

BY

I. B. TURKSEN

A. G. HOLZMAN

In the report "Boolean Isomorphic Structures in an Educational Scheduling Model", by Holzman and Turksen, basic concepts of the scheduling system were investigated and a methodology was developed to generate a feasible solution space which does not violate any of the constraints of the problem.

The feasible solution space of the ten variable problem showed that there were many possible combinations for time assignments according to some criterion. In this research the criterion used is to minimize the number of time periods required to satisfy all the constraints imposed upon the scheduling system.

A Relations Matrix is developed from the constraints of the system. The selection of the minimum number of time period vectors is obtained with the specifications of a policy and Bellman's principle of optimality. The minimality of the number of vectors varies as a function of constraints and policy.

It is desirable to review the hypothetical 10-variable problem discussed in "Boolean Isomorphic Structures in an Educational Scheduling Model." The total feasible solution of this problem consists of 102 vectors (feasible configurations). The number of vectors is large in comparison with the number of variables, and many alternative selections exist . Each alternative may or may not have the minimum number of vectors, but they will all satisfy the constraints indicated. An objective attempt might be the search for the minimum number of vectors which will satisfy the concept of a solution. One such alternative is the following:

	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1
Vector 1	0	0	0	1	0	0	0	0	0	0
Vector 2	0	1	0	0	0	0	0	0	0	0
Vector 3	0	0	1	0	1	0	0	0	0	0
Vector 4	1	0	0	0	1	1	0	0	1	1

The ones of each vector indicate the variables that can be assigned to the same independent time period. For instance. Vector 4 above indicates that: A_{10}, A_6, A_5, A_2, A_1 can be assigned to the same independent time period and the constraints of the problem would not be violated. The constraints of the problem are satisfied when an independent time period for each vector is assigned.

A closer study of the above vectors reveals the phenomena of the relationships of variables with respect to each other. These phenomena were implicitly mentioned when the variables and the functions of a scheduling system were defined in the previous paper.

The zeros and ones of each vector of the feasible solution space simply indicate the relations of a given variable with respect to all others.

	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1
Vector 1	0	0	0	1	0	0	0	0	0	0
Vector 4	1	0	0	0	1	1	0	0	1	1

Vector 1 implicitly indicates that Variable A_7 cannot be given at the same time period with any other variable of the problem. Put another way, when A_7 is assigned to a time period no other variable can be assigned to that time period. Vector 4 indicates the relation that variables A_{10}, A_6, A_5, A_2, A_1 can be assigned to the same time period.

But A_9, A_8, A_7, A_4, A_3 must not be assigned to the time period which was assigned to A_{10}, A_6, A_5, A_2, A_1.

Further study of the relations phenomena leads to the construction of the Relations Matrix.

NEW CONCEPTS

A Relations Matrix is a symmetric square matrix. The row and column headings of this matrix are the names of the variables of the scheduling system, i.e. the course sections. If $R=[r(i,j)]$ is the relations matrix of order nxn, where n=number of variables, then the elements of $[r(i,j)]$ are either 1 or 0. The diagonal of $R=(1, 1, 1, \ldots, 1)$. This indicates that $r(i,i)=1$ for all i, i.e., a variable can have no constraint with itself,- no constraint of the form $A_i \cdot A_i=0$ is meaningful. For each row i, the '1' elements in the row indicate the variables that might be scheduled when the i^{th} variable is scheduled. The interpretation made for the '0's and '1's of each row are in reality basic for the construction of the Relations Matrix.

Figure 1 is the Relations Matrix for the 10-variable problem specified in the previous report. The zeros are established with respect to the constraints given in the construction of the problem.

$$\begin{array}{c|cccccccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\
\hline
A_1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
A_2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
A_3 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
A_4 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
A_5 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
A_6 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
A_7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
A_8 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
A_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
A_{10} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1
\end{array}$$

Figure 1

Construction of Relations Matrix

1.) Establish an nxn matrix.

2.) Make each element of the matrix '1'.

3.) For each constraint $Ai \cdot Aj = 0$, make $r(i,j)=0$, and $r(j,i)=0$ in the relations matrix.

When the first two steps are executed, the state of the Relations Matrix has a special meaning. This special state is the outcome of non-constrained conditions. In class scheduling systems this implies that all course sections can be offered at a single time period. This state is also the indication of unlimited physical plant facilities, unlimited teaching staff, and unlimited students with a single course section request. When the constraints begin to act upon the non-constrained state, the zeros of the Relations Matrix come into existence.

Example: $n=4$

Constraints: $A_4 \cdot A_3=0$

$A_3 \cdot A_2=0$

$A_4 \cdot A_1=0$

1.) Establish a 4x4 matrix.

2.) Make each element of the matrix '1'.

$$\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline A_1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 \\ A_3 & 1 & 1 & 1 & 1 \\ A_4 & 1 & 1 & 1 & 1 \end{array}$$

3.) For each constraint $Ai \cdot Aj =0$, make $r(i,j)=0$, and $r(j,i) =0$.

$A_4 \cdot A_3=0$

$r(4,3)=0$

$r(3,4)=0$

$$\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline A_1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 \\ A_3 & 1 & 1 & 1 & 0 \\ A_4 & 1 & 1 & 0 & 1 \end{array}$$

$A_3 \cdot A_2=0$

$r(3,2)=0$

$r(2,3)=0$

$$\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline A_1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 0 & 1 \\ A_3 & 1 & 0 & 1 & 0 \\ A_4 & 1 & 1 & 0 & 1 \end{array}$$

$$A_4 \cdot A_1=0$$

$$r(4,1)=0$$

$$r(1,4)=0$$

$$\begin{array}{c|cccc} & A_1 & A_2 & A_3 & A_4 \\ \hline A_1 & 1 & 1 & 1 & 0 \\ A_2 & 1 & 1 & 0 & 1 \\ A_3 & 1 & 0 & 1 & 0 \\ A_4 & 0 & 1 & 0 & 1 \end{array}$$

It should be indicated here that Class I constraints ($Ai \cdot Aj=0$) are directly applicable to the construction of the Relations Matrix. The other constraints, Class II - where one course has a single section and the other course has multi-sections, and Class III - where both courses have multi-sections, can be reduced to Class I constraints.*

INTERPRETATION OF RELATIONS MATRIX

The structuring of a Relations Matrix gives a particular meaning to each row of this matrix. It was indicated that $r(i,i)=1$ for i, and that $r(i,j)=0$, if and only if a constraint of the form $Ai \cdot Aj=0$ exists for a particular j, for $1 \leq j \leq n$, and $j \neq i$. Therefore the '0's in row i indicate all the variables that cannot be assigned to the time period that variable i is assigned. Furthermore, the total number of '0's is the total number of constraints applied to variable i. Row i has no explicit information about total constraints applied to any other variable. Note, however, that it has implicit information that does appear in other rows for single constraints because of the symmetric nature of the Relations Matrix.

---

* For definition of Class I, Class II, and Class III constraints, refer to Boolean Isomorphic Structures in an Educational Scheduling Model by Holzman and Turksen.

The '1' entries of the Relations Matrix also have further implications. If r(i,j)=1 in row i, then A_j may be scheduled with A_i. Because of the symmetry, r(j,i)=1 in the j^{th} row. By construction, r(i,i)=1 in the i^{th} row and r(j,j)=1 in the j^{th} row. The Boolean AND operation is performed on the i^{th} row vector and the j^{th} row vector.

The Resultant Vector RV(k) =r(i,k) AND r(j,k), for k=1, 2,..., n.

Example:

				i		j			
i^{th} row	1	0	......	1	...0	1	1	...	1
j_{th} row	1	1	...	1	...1	1	0	...	1
Resultant Vector	1	0	...	1	...0	1	0	...	1

If r(i,j)=1, the minimum information that can be obtained is RV (i)=r(i,i) AND r(j,i)=1, and RV (j)=r(i,j) AND r(j,j)=1. Therefore if r(i,j)=1, we are assured that at least two entries of the resultant vector, namely the i^{th} and the j^{th}, will be equal to '1'. If for some k, r(i,k)=0, regardless of the value of r(j,k), then RV (k)=0 as a result of the Boolean operator AND. If, for some other k, r(j,k)=0, then RV (k)=0, regardless of the value of r(i,k). The information contained in the resultant vector is significant. The resultant vector will have zeros everywhere the i^{th} row has zeros and it will have zeros everywhere the j^{th} row has zeros. The significance comes from the fact that the resultant vector will indicate all the variables that cannot be scheduled when the i^{th} and the j^{th} variables are scheduled.

Furthermore, the '1's' of the resultant vector indicate other variables that may be scheduled at the same time as the i^{th} and j^{th} variables.

The binary and commutative properties of the Boolean operator AND enable us to obtain the influencing constraints of all other variables by an iterative procedure. For instance, once $RV(k) = r(i;k)$ AND $r(j;k)$ is established, and if $RV(p) = 1$, for $1 \leq k = p \leq n$, with $p \neq i$, $p \neq j$, then we construct $RV'(k) = RV(k)$ AND $r(p;k)$, for $k = 1, 2, \ldots, n$. The new resultant vector will now contain information about the relationship of variables i, j, p, to the remaining variables. The '0's in $RV'(k)$ indicate the variables that cannot be scheduled when the i^{th}, j^{th} and p^{th} variables are scheduled together.

Determination of Independent Time Periods from the Relations Matrix

Algorithm: (Refer to flow diagram in Figure 2)

1. Select an i^{th} row with respect to a given policy.
2. Does the i^{th} row have a non-zero element, other than $r(i,i)$?
   If yes, go to (3)
   If no, schedule i^{th} course section to this independent time period by itself, and go to (9).
3. Select a non-zero element of the i^{th} row with respect to given policy. Let it be in the j^{th} column. Thus, $r(i,j)=1$.
4. Determine: $RV'(k) = r(i;k)$ AND $r(j;k)$, for $k = 1, 2, \ldots, n$
5. Make $RV(k) = RV'(k)$
6. Have all non-zero elements of $RV(k)$ been scheduled?
   If yes, go to (8)
   If no, go to (7)

7. Select a non-zero element of RV (k) with respect to given policy. Let it be in the p^{th} column.

   Determine: RV' (k) = RV (k) AND r (p;k), for k = 1,2,..., n

   Go to (5).

8. Assign an independent time period to the course sections which have a non-zero element in RV (k). All of these course sections can be scheduled in the same time period without conflicts.

9. Have all course sections been scheduled?

   If yes, stop.

   If no, go to (1), selecting a new i^{th} row with respect to given policy.

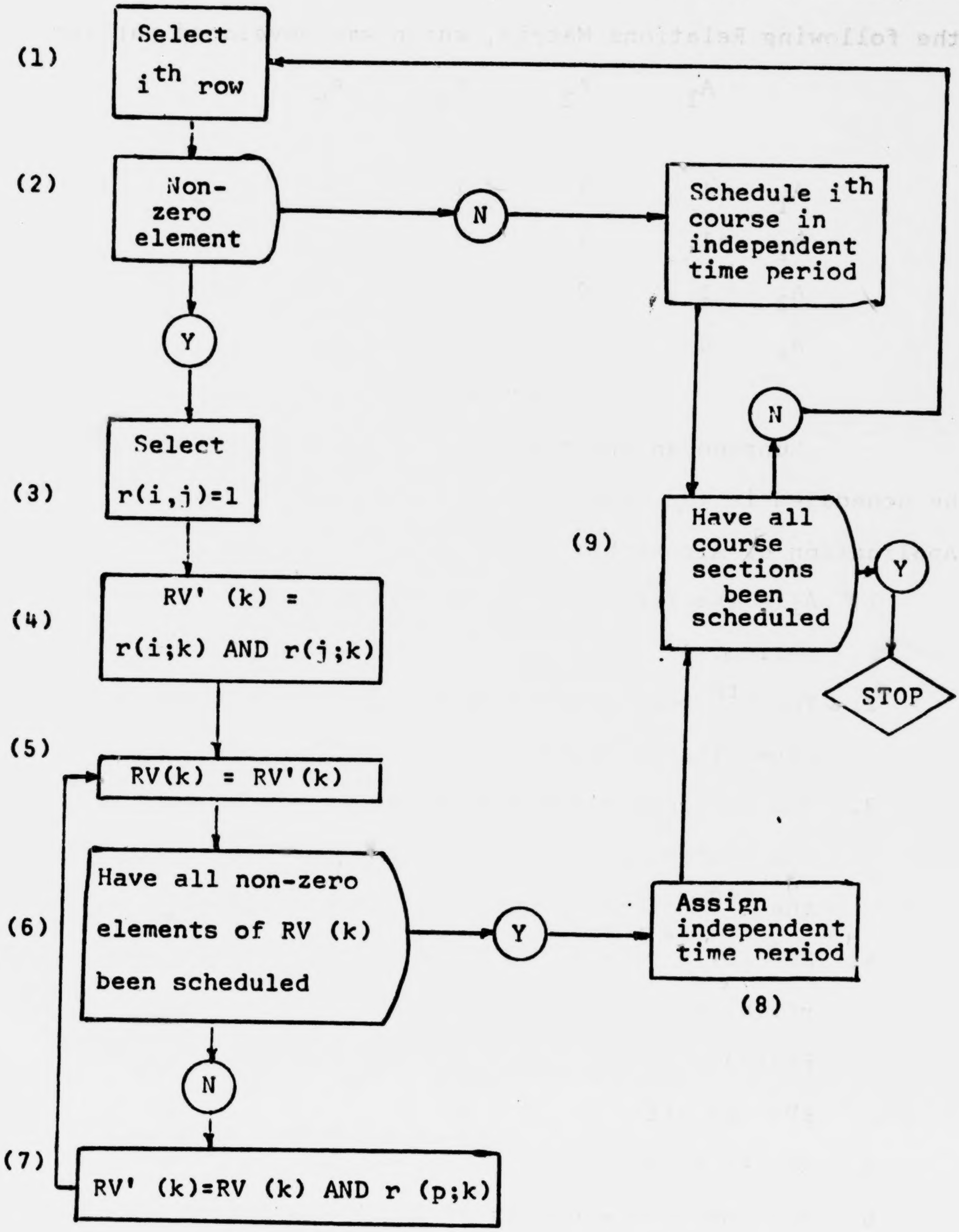

(1)
Select
i th row
(2)
Non-zero element
N
Schedule i th course in independent time period
Y
N
(3)
Select
r(i,j)=1
(9)
Have all course sections been scheduled
Y
(4)
RV' (k) =
r(i;k) AND r(j;k)
STOP
(5)
RV(k) = RV'(k)
(6)
Have all non-zero elements of RV (k) been scheduled
Y
Assign independent time period
(8)
N
(7)
RV' (k)=RV (k) AND r (p;k)

Figure 2

Example: The algorithm will now be applied to determine the assignment of course sections to independent time periods for the following Relations Matrix, which was developed earlier.

$$\begin{array}{ccccc} & A_1 & A_2 & A_3 & A_4 \\ A_1 & 1 & 1 & 1 & 0 \\ A_2 & 1 & 1 & 0 & 1 \\ A_3 & 1 & 0 & 1 & 0 \\ A_4 & 0 & 1 & 0 & 1 \end{array}$$

Figure 3

Suppose an arbitrary policy states that variables should be scheduled in the order of A_1, A_2, A_3, A_4.

Application of Algorithm:

1. A_1 is the first row to be selected with respect to given policy.
2. The i^{th} row, denoted by A_1, does have a non-zero element, other than r (1,1).
3. The non-zero element selected in row A_1 with respect to the stated policy is A_2. Since the i^{th} row is row 1, and the j^{th} column is column 2, then r(1,2)=1.
4. RV' (k) = r(1;k) AND r(2;k), for k = 1,2,...,n.

   $r(1;k) = (1 \quad 1 \quad 1 \quad 0)$

   $r(2;k) = \underline{(1 \quad 1 \quad 0 \quad 1)}$

   $RV'(k) = (1 \quad 1 \quad 0 \quad 0)$
5. $RV(k) = RV'(k)$
6. All non-zero elements of RV (k) have been scheduled.
8. Course section A_1 and A_2 are assigned to an independent time period.

9. All course sections have not been scheduled. Start new cycle denoted by primes as follows.

1.' A_3 is the new i^{th} row to be selected with respect to given policy.

2.' A_3 does have a non-zero element, other than r(3,3).

3.' The non-zero element selected in row A_3 with respect to the stated policy is A_1. Since the i^{th} row is row 3, and the i^{th} column is column 1, then r(3,1) =1.

4.' RV'(k) =r(3;k) AND r(1;k), for k=1,2, ..., n

r(3;k) = (1 0 1 0)
r(1;k) =$\underline{(1 \quad 1 \quad 1 \quad 0)}$
RV' (k)=(1 0 1 0)

5.' RV(k) = RV' (k)

6.' All non-zero elements of RV(k) have been scheduled.

8.' Course sections A_3 and A_1 are assigned to an independent time period.

9.' All course sections have not been scheduled. Start new cycle, denoted by double primes.

1." A_4 is the new i^{th} row to be selected with respect to given policy.

2." A_4 does have a non-zero element, other than r(4,4).

3." The non -zero element selected in row A_4 with respect to the stated policy is A_2. Since the i^{th} row is row 4, and the j^{th} column is column 2, then r(4,2)=1.

4." RV' (k)=r(4;k) AND r(4;k) , for k = 1, 2,..., n.

$r(4;k) = (0 \quad 1 \quad 0 \quad 1)$

$r(2;k) = \underline{(1 \quad 1 \quad 0 \quad 1)}$

$RV'(k) = (0 \quad 1 \quad 0 \quad 1)$

5." $RV(k) = RV'(k)$

6." All non-zero elements of $RV(k)$ have been scheduled.

8." Course sections A_4 and A_2 are assigned to an independent time period.

9." All course sections have been scheduled. Stop.

Therefore, three independent time periods which satisfy the constraints of the problem are:

$$\begin{array}{c c} & \begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] \end{array} = S_1$$

It is interesting to note that if the policy were to take A_1, A_3, A_4, A_2, the independent time periods would be

$$\begin{array}{c c} & \begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \left[\begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] \end{array} = S_2$$

and that two independent time periods would be enough. This simple example shows that the minimality of the independent time periods is a function not only of the constraints but also of the way a given policy has been structured.

A POLICY PROCEDURE

Two distinct policies produced two distinct alternatives, S_1 and S_2 for the previous example. Let us study Matrix S_1.

The rows of S_1 were obtained with respect to the policy which stated that the variables should be scheduled in the given order of A_1, A_2, A_3, A_4. The vector (1 1 0 0) was obtained as the first row of S_1, the vector (1 0 1 0) was obtained as the second row, and finally, the vector (0 1 0 1) was obtained as the third row.

It is obvious that A_1 is scheduled in the first row, but it is re-scheduled in the second row. A similar situation exists for A_2 in rows 1 and 3. In general, it may be concluded that a subset of variables may be re-scheduled in a number of independent time periods. This in turn leads to the occurrence of sub-alternatives within an alternative of a given policy. If it is desired that a solution for a policy procedure be one of the sub-alternatives and thereby eliminate the further complexities of interpretation, Bellman's principle of optimality might be used. Bellman's principle of optimality states:

> An optimal policy has the property that, whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with respect to the state resulting from the first decision.

In scheduling systems an optimality procedure for a given policy should be stated in some manner. One such optimality procedure would be to eliminate the effects of previously scheduled variables before the construction of a new schedule vector is started. Above it was found that the first vector of S_1 is (1 1 0 0). At this point the policy states that the next variable to be scheduled is A_3. Therefore, the third row of the Relations Matrix should be processed: $r(3;j) = (1\ \ 0\ \ 1\ \ 0)$.

Now, before $r(3;j)$ is processed, the effect of previously scheduled variables should be eliminated from this vector. Since A_1 and A_2 have already been assigned to an independent time period, they should be eliminated from further consideration. When procedure of elimination is applied, the vector $r(3;j) = (0\ 0\ 1\ 0)$ is obtained. When this procedure is applied to the fourth row, the vector $r(4;j) = (0\ 0\ 0\ 1)$ is obtained.

It will be necessary to define the characteristics of procedure of optimality on scheduling systems. The procedure of optimality, in a sense, frees the incoming vector from the restrictions of the previously assigned variables. Only one additional step is required in the algorithm developed previously.

Step 1-A - Make all elements of the i^{th} row equal to zero for those variables already assigned to an independent time period.

Given the following Relations Matrix, with the policy stated as A_1, A_2, A_3, A_4, A_5, the independent time periods will be determined (1) without procedure of optimality and (2) with procedure of optimality.

$$\begin{array}{c|ccccc} & A_1 & A_2 & A_3 & A_4 & A_5 \\ A_1 & 1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 0 & 1 & 0 \\ A_3 & 1 & 0 & 1 & 0 & 1 \\ A_4 & 0 & 1 & 0 & 1 & 0 \\ A_5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Figure 4

(1) Without procedure of optimality - Four independent time periods are obtained as shown in Matrix S_3.

$$\begin{array}{cccccc} & A_1 & A_2 & A_3 & A_4 & A_5 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \end{array} = S_3$$

(2) With procedure of optimality - The independent time periods will now be developed from the expanded algorithm, which includes Step 1-A: Make all elements of the i^{th} row equal to zero for those variables already assigned to an independent time period.

Refer to the Relations Matrix in Figure 4.

1. A_1 is the first row to be selected with respect to given policy.

1A. Since no course section has been scheduled as yet, no change is made in the elements of the first row.

2. The i^{th} row A_1 does have a non-zero element other than $r(1,1)$.

3. The non-zero element selected in row A_1 with respect to the stated policy is A_2. Thus $r(1,2) = 1$

4. $RV'(k) = r(1;k)$ AND $r(2;k)$

   $= (1\ 1\ 0\ 0\ 0)$

5. $RV(k) = RV'(k)$

6. All non-zero elements of $RV(k)$ have been scheduled.

8. Course sections A_1 and A_2 are assigned to an independent time period.

9. All course sections have not been scheduled. Start new cycle.

1.' A_3 is the i^{th} row to be selected with respect to given policy.

1A.' Make r(3,1)=0, and r(3,2)=0, since course sections A_1 and A_2 have already been scheduled.

2.' A_3 does have a non-zero element, other than r(3,3).

3.' The non-zero element selected in row A_3 is A_5. Thus, r(3,5) =1

4.' RV'(k) = r(3;k) AND r(5;k)

5.' RV(k) = RV'(k)

= (0 0 1 0 1)

6.' All non-zero elements of RV(k) have been scheduled.

8.' Course sections A_3 and A_5 are assigned to an independent time period.

9.' All course sections have not been scheduled. Start new cycle.

1." A_4 is the new i^{th} row to be selected.

1A." Make r(4,1)=0, r(4,2)=0, r(4,3)=0 and r(4,5)=0 since course sections A_1, A_2, A_3, A_5 have already been scheduled.

2." A_4 does not have a non-zero element, other than r(4,4). Therefore, A_4 is scheduled in an independent time period by itself to give the vector (0 0 0 1 0)

9." All course sections have been scheduled. Stop. Thus, with the procedure of optimality, three independent time periods are obtained as shown in Matrix S_4.

$$\begin{array}{c|ccccc} & A_1 & A_2 & A_3 & A_4 & A_5 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 \\ 3 & 0 & 0 & 0 & 1 & 0 \end{array} = S_4$$

It is seen that, even though the procedure of optimality appears to be severe, it in fact relaxes the constraints that may otherwise be imposed on a vector.

SUMMARY

A study of the vector structure in the previous research on Boolean Isomorphic Structures revealed the phenomena of the relationship of variables. This resulted in the generation of a Relations Matrix from the constraints of the scheduling system. The minimum number of time periods is obtained from the specification of a policy and the principle of optimality.

DECOMPOSED MODEL
OF AN
EDUCATIONAL SCHEDULING SYSTEM
FOR
OPTIMAL MASTER SCHEDULE CYCLE

BY

I. B. TURKSEN

A. G. HOLZMAN

BASIC CONCEPTS

In this study we conceive a model of the educational scheduling system. Before the system can be analyzed, the elements of this system must be defined concomitant with an identification of its capabilities and functional performance expected. Capabilities would include courses that can be offered, faculty availability, and physical plant capacity. With these capabilities, the system must provide the service required by its students. This is to be translated into the development of a non-conflicting schedule of classes for each student.

Educational scheduling activities may be decomposed into phases. This decomposition follows for our structure of the model.

Phase I. A. Determination of Course Sections to be Offered.

B. Administrative Changes.

Phase II. Assignment of Course Sections to Faculty, Students and Physical Plant.

Phase III. Development of the Constraint Structures Imposed on the System.

Phase IV. Establishment of Priorities Based on the System Constraints and the Course Specifications.

Phase V. Construction of the Master Schedule to Satisfy the Conditions Established in Phase I through IV.

The basic structures that are used in the development of these five phases will now be defined. Some of these structures are commonly known, while others are defined for the first time.

The time interval for which the schedule is prepared is called cycle. Ordinarily, a schedule cycle is one week, which consists of either five days or six days, depending upon whether Saturday classes are allowed in the educational institution. However, in this research the length of the cycle is determined by the optimal number of schedule days required to satisfy the constraints in the scheduling system.

Each schedule cycle is divided into *days* with each day divided into *sessions*.Each day consists of two sessions, morning session and afternoon session. Each session is divided into a number of *periods*.

Now we will develop a precise definition for a period as used in this study. Two course time increments a and b are congruent modulo m only if the difference (a-b) lies in the set of all multiples of m.

Suppose the time increment for a meeting of Mathematics 24 is 60 minutes, and the time increment for a meeting of Chemistry 12 lab is 120 minutes. Then we say $120 \equiv 60$ (modulo 60). Also $120 \equiv 60$ (modulo 30).

A period is defined as the largest time increment modulo m so that all time increments are congruent.

For example, if in the courses to be scheduled, some of the courses meet 60 minutes per day, some 120 minutes per day, and others 180 minutes per day, then the largest modulo m so that all course time increments are congruent is 60 minutes.

$180 \equiv 120$ (modulo 60)

$120 \equiv 60$ (modulo 60)

Therefore, a *period* in this problem would be defined as 60 minutes.

If in the courses to be scheduled, some meet 60 minutes per day, some meet 90 minutes per day, and others meet 120 minutes per day, then the largest modulo m for the congruence is 30 minutes, and the period is 30 minutes.

$$120 \equiv 90 \pmod{30}$$

$$90 \equiv 60 \pmod{30}$$

The number of periods for the morning session may be different from the number for the afternoon session, but the time interval for a period must be the same for each session.

Since usually not all courses meet the same number of hours per day, it is necessary to establish a period multiple, K, for each course, where K is a positive integer equal to or greater than 1. As an example, if a period is one hour, and the course meets 2 hours per day, then K =2. For reasons to be recognized later in this presentation, we will call a period a course unit when it is used in this context. Thus, for the example above where the period is one hour, and the course meets 2 hours per day, we say the course meets 2 units per day.

PHASE I

A. DETERMINATION OF COURSE SECTIONS TO BE OFFERED

A one to one onto mapping is established between the Course Curriculum Identifiers and the set of positive integers P, where P is the set of numbers from 1 to 9999. This mapping translates the Course Curriculum Identifiers to μ-Identifiers as shown in Figure 1, to facilitate computer use.

Course Curriculum Identifier	μ-Identifiers
Mathematics 24	$\mu_1 = 1$
Mathematics 100	$\mu_2 = 2$
⋮	⋮
Physics 96	$\mu_k = k$
⋮	⋮
Economics 6	$\mu_n = n$

Figure 1

The list of μ-Identifiers may also be called the list of courses that may be offered at the educational institution. The order in which the Course Curriculum Identifiers are listed is not important. The correspondence is between the positional place of the Curriculum Identifiers and the subset of positive integers, 1 - 9999. For example, in Figure 1, Mathematics 24 could be listed in position 2,3,..., or n, with the μ-Identifier changing to 2,3,..., or n, to correspond with the position in which the course is listed.

The Faculty Availability Vector, FAV, is a vector whose elements are the number of faculty members available to teach a course. A positional one to one mapping also exists between the μ-Identifiers and the elements of the Faculty Availability Vector, as shown in Figure 2.

μ-Identifiers	Faculty Availability Vector - FAV
$\mu_1 = 1$	$f_1 = 10$
$\mu_2 = 2$	$f_2 = 2$
$\vdots$	$\vdots$
$\mu_k = k$	$f_k = 1$
$\vdots$	$\vdots$
$\mu_n = n$	$f_n = 5$

Figure 2

In Figure 2, f_1 is the number of faculty members available to teach course μ_1, and f_k is the number of faculty members available to teach course μ_k. If a faculty member can teach more than one section of a course μ_k, then the total number of times that he is to teach course μ_k must be counted.

The set of courses a student is planning to take is called a student request set. A student request set consists of the student identification number and the μ-Identifiers of the courses he is planning to take. The kth student request set is shown as; s_k: j...i...s; where s_k is the student identification and i,..., j,..., s, μ-Identifiers of the courses requested.

A counter is associated with each μ-Identifier. The Counter Vector has the associated counters for its elements, as seen in Figure 3.

μ-Identifiers	Counter Vector
$\mu_1 = 1$	c_1
$\mu_2 = 2$	c_2
$\vdots$	$\vdots$
$\mu_k = k$	c_k
$\vdots$	$\vdots$
$\mu_n = n$	c_n

Figure 3

The contents of each counter c_k is the number of student requests for the course μ_k.

Usually, an educational institution requires a minimum number of student requests for a course to be offered. The vector containing elements which indicate the minimum number of requests required for the offering of a course is called the Minimum Requirement Vector. The usual positional correspondence is used between the μ-Identifiers and the Minimum Requirement Vector, as shown in Figure 4.

μ-Identifiers	Minimum Requirement Vector
$\mu_1 = 1$	$mir_1 = 20$
$\mu_2 = 2$	$mir_2 = 15$
$\vdots$	$\vdots$
$\mu_k = k$	$mir_k = 30$
$\vdots$	$\vdots$
$\mu_n = n$	$mir_n = 75$

Figure 4

For example, $mir_2 = 15$ indicates that the minimum number of students required to offer course 2 is 15.

Usually, it is also desired to control the maximum number of students to be placed in a course. If there are more requests than the maximum number that may be allowed, then the course is decomposed into more than one section.

The vector which indicates the maximum number of students allowed in a course section is called the Maximum Requirement Vector. The usual one to one correspondence exists as shown in Figure 5.

μ-Identifiers	Maximum Requirement Vector
$\mu_1 = 1$	$mxr_1 = 25$
$\mu_2 = 2$	$mxr_2 = 30$
⋮	⋮
$\mu_k = k$	$mxr_k = 85$
⋮	⋮
$\mu_n = n$	$mxr_n = 85$

Figure 5

For example, $mxr_2 = 30$ indicates that the maximum number of students permitted in course 2 is 30.

It was pointed out above that the student demand may not warrant offering a course; on the other hand, more than one section of a course may be signalled by the student demand for the particular course. During this Phase I, the number of sections that need to be offered for course μ_k is determined from a comparison of the student requests in the Counter Vector with the Minimum Requirement Vector and the Maximum Requirement Vector.

Two vectors, called the Need-Vector and the Unsatisfied Requests Vector, are generated. An element n_k of the Need Vector is the number of sections needed to service the student requests for the course μ_k. An element r_k of the Unsatisfied Requests Vector is the number of requests that will be unsatisfied if n_k remains as the number of sections of the course μ_k. Refer to Figure 6.

μ-Identifiers	Need-Vector	The Unsatisfied Requests Vector
$\mu_1 = 1$	n_1	r_1
$\mu_2 = 2$	n_2	r_2
$\vdots$	$\vdots$	$\vdots$
$\mu_k = k$	n_k	r_k
$\vdots$	$\vdots$	$\vdots$
$\mu_n = n$	n_n	r_n

Figure 6

For any μ-Identifier, μ_k, in Figure 6, n_k and r_k are determined as follows:

a) if $c_k < mir_k$, then $n_k = 0$ and $r_k = c_k$. The c_k, which is obtained from the counter vector, indicates the number of students requesting μ_k.

b) If $mir_k \leq c_k \leq mxr_k$, then $n_k = 1$, $r_k = 0$.

c) If $c_k > mxr_k$, then n_k is found by Euclid's division Algorithm, $c_k = n_k \cdot mxr + r_k$ where r_k is the left over, ie; the number of unsatisfied requests.

The Need-Vector, N, is compared against the Faculty Availability Vector, FAV. A vector CRV, called the Comparative Result Vector, is generated. Refer to Figure 7.

μ-Identifiers	Comparative Result Vector
μ_1	crv_1
μ_2	crv_2
.	.
.	.
.	.
μ_k	crv_k
.	.
.	.
.	.
μ_n	crv_n

Figure 7

For a course whose μ-Identifier is k, with n_k and f_k being the number of sections needed and the number of faculty members available respectively,

a) If $n_k = f_k$, then n_k sections can be offered, and $crv_k = 0$.

b) If $n_k < f_k$, then n_k sections can be offered and $f_k - n$ faculty members will be available for other assignments and $crv_k = n_k - f_k$.

c) If $n_k > f_k$, then f_k course sections can be offered and $n_k - f_k$ number of sections cannot be offered and $crv_k = n_k - f_k$.

The activities of Phase I A may be summarized in three steps.

1. Counter Vector: The student request sets are tallied. The number of requests for each course is determined, i.e. $c_1, c_2, \ldots, c_n$ are found for μ_k's, $k = 1, 2, \ldots, n$.

2. Need Vector and Unsatisfied Requests Vector: The counter vector elements are analyzed with respect to the Minimum Requirement Vector and the Maximum Requirement Vector. The elements of the Need Vector, $n_1, n_2, \ldots, n_n$, and the elements of the Unsatisfied Requests Vector, $r_1, r_2, \ldots, r_n$ are calculated.

3. Comparative Result Vector: The Need Vector, N, is compared against the Faculty Availability Vector, FAV. The sign of the difference $n_k - f_k$ is interpreted with specific meaning. If crv_k is negative, then faculty is available for assignment to courses other than μ_k. If crv_k is positive, then more faculty is needed to satisfy the student demands for μ_k.

B. ADMINISTRATIVE CHANGES

The information obtained in Phase I A could, at best, portray the environment of the educational scheduling system. The Minimum Requirement Vector, the Maximum Requirement Vector, and the Faculty Availability Vector are a set of estimators and should be subject to adjustments.

For example, if for μ_k, $mir_k = 10$, $mxr_k = 20$, $c_k = 9$, and $f_k = 2$, then $n_k = 0$, $r_k = 9$ and $crv_k = 2$ at the end of Phase I A. It may seem that a section of μ_k should be offered even though the requests are below the Minimum Requirement. If this is the case, then $n_k = 0$ should be made $n_k = 1$ and $mxr_k = 20$.

If, for another μ_k, $mir_k = 12$, $mxr_k = 20$, $c_k = 55$, and $f_k = 2$, then $n_k = 2$, $r_k = 15$, and $crv_k = 0$. If it is possible to assign another faculty member to μ_k, then the remaining 15 requests would be satisfied by changing $n_k = 2$ to $n_k = 3$ and $mxr_k = 20$.

If, for still another μ_k, $mir_k=15$, $mxr_k=20$, $c_k=44$, and $f_k=2$, then $n_k=2$, $r_k=4$ and $crv_k=0$. If it is possible to change $mxr_k=22$, then unsatisfied requests, $r_k=4$, can be absorbed by the existing 2 sections. Thus, a change in mxr_k for any μ_k may be in question.

It is suggested that the specific changes be made manually to preserve desirable humanistic features of such adjustments.

The above considerations lead to the generation of a new listing. An A-Identifier list is generated for the course sections that are to be offered. The A-Identifiers are generated by attaching a two digit number to the right of the μ-Identifiers.

For example, if three courses, μ_4, μ_6, μ_9, are to be offered, with μ_4 requiring 3 sections, μ_6 requiring 1 section, and μ_9 requiring 2 sections, then the following correspondence is established between the μ-Identifiers and the A-Identifiers.

μ-Identifiers	A-Identifiers
$\mu_4 = 4$	$A_1^* = 4\ 01$
	$A_2 = 4\ 02$
	$A_3 = 4\ 03$
$\mu_6 = 6$	$A_4 = 6\ 01$
$\mu_9 = 9$	$A_5 = 9\ 01$
	$A_6 = 9\ 02$

Figure 8

*Capital "A" instead of lower case "a" will be used to denote the elements of the A vector to be consistent with the course designation used by authors in another paper.

If n_k is equal to zero, then μ_k is not offered.

Thus, it can be seen from Figure 8 that course section A_3 is one of the sections for the course μ_4. From this point on, course section designation, A_1, A_2,... A_n, will be utilized in this discussion. The course sections indicated by the A -Identifiers may be considered as a string of elements which are to be offered to the students and to be taught by a faculty member.

Each A_k of the vector A is a course section of some course which can be determined by the correspondence between the A-Identifiers and the μ -Identifiers for courses to be offered. The correspondence from μ -Identifiers to A-Identifiers is established primarily (1) to distinguish between specific sections for each course; and (2) to generate a sequential listing of course sections for later use.

Very concisely, we can say that Phase I is primarily concerned with student demand and faculty availability. No assignments are made in Phase I.

Flow Charts I and II depict the activities of Phase I. Refer to Figures 9 and 10.

FLOW CHART I

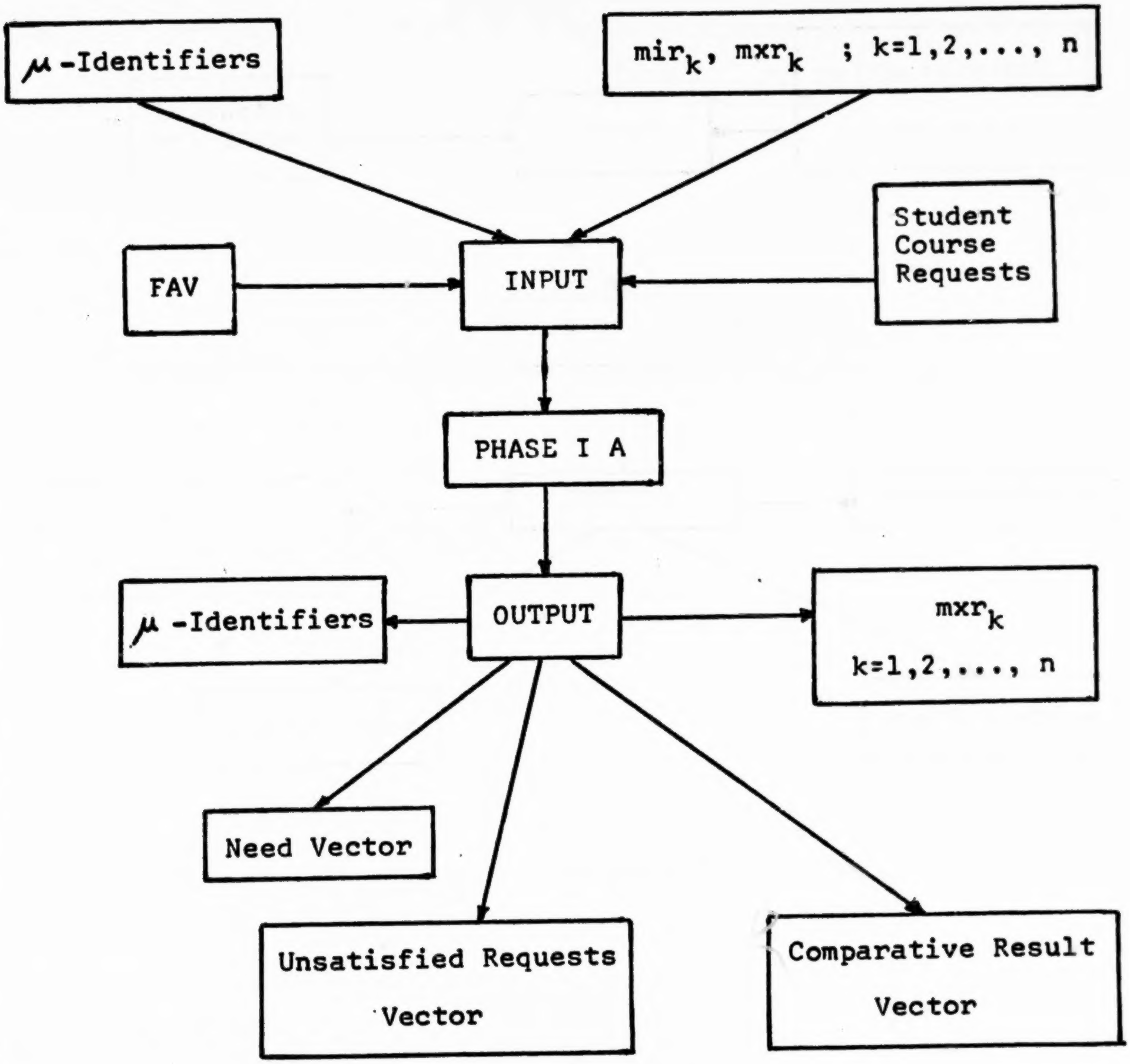

μ-Identifiers
mir_k, mxr_k ; k=1,2,..., n
FAV
INPUT
Student Course Requests
PHASE I A
μ-Identifiers
OUTPUT
mxr_k k=1,2,..., n
Need Vector
Unsatisfied Requests Vector
Comparative Result Vector

Figure 9

FLOW CHART II

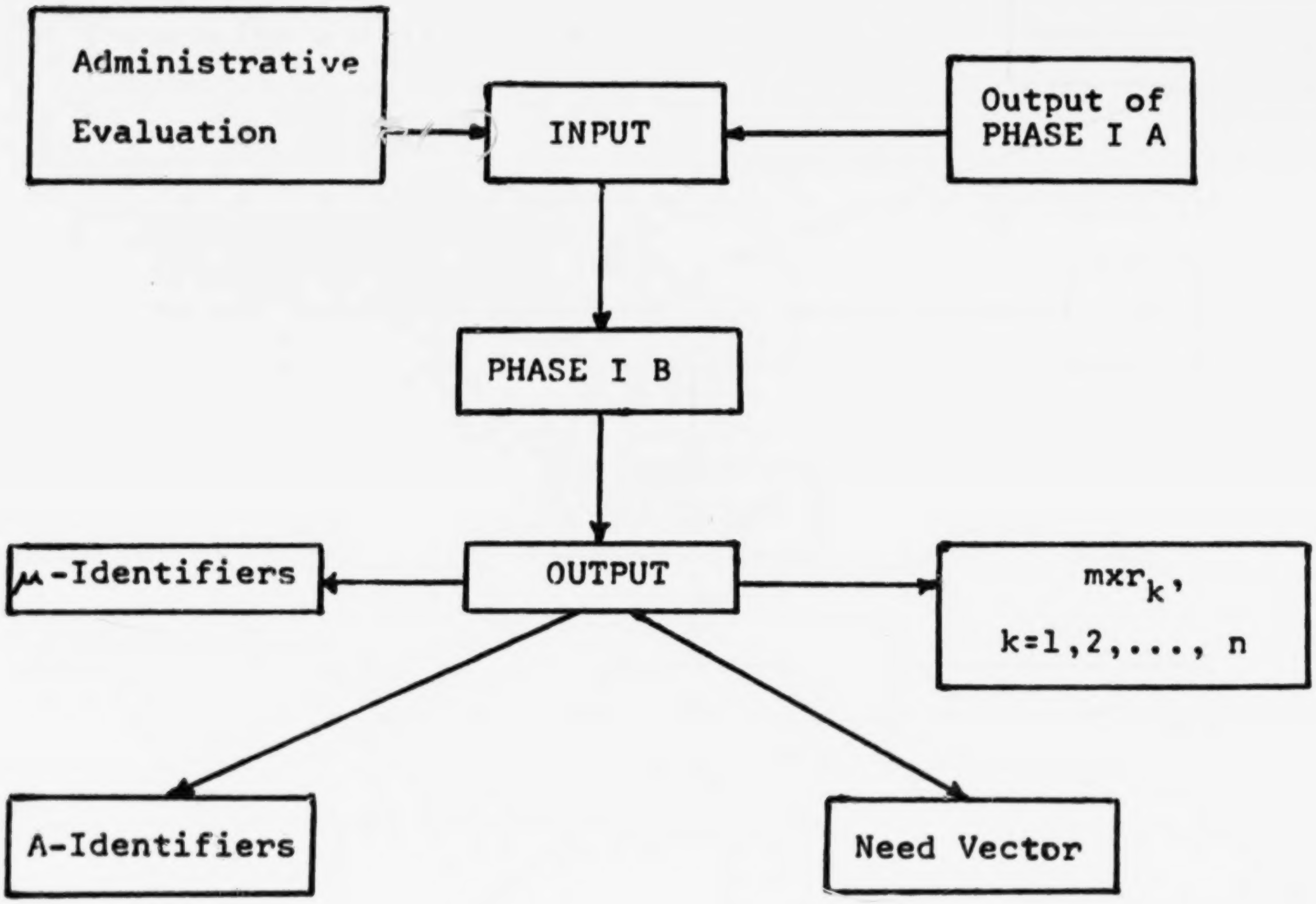

Administrative Evaluation
INPUT
Output of PHASE I A
PHASE I B
OUTPUT
μ-Identifiers
mxr_k, k=1,2,..., n
A-Identifiers
Need Vector

Figure 10

PHASE II

ASSIGNMENT OF COURSE SECTIONS TO FACULTY, STUDENTS, AND PHYSICAL PLANT

In Phase II, the course sections are assigned to specific faculty members, students are assigned to the specific course sections, and for the critical facilities, such as lecture auditoriums and lab facilities, the courses that are to meet in these facilities are indicated. For ordinary recitation and lecture meetings, not requiring special facilities, only the total number of available classrooms must be known.

It is suggested that the specific faculty assignments be made manually to preserve desirable humanistic features of such arrangements. The critical facility assignments are also made manually.

Course sections are assigned to individual students during this phase. That is, the specific sections are assigned to the students for the courses which they have requested.

Where multiple sections exist, assignment of a particular section to a student may be accomplished in various ways. For example, students may be assigned on the basis of their scholastic ability to insure more homogeneous grouping in the sections. Faculty preference of a student may also be used as a basis for section assignments. To insure interaction of students from various departments or schools in a university, the policy may be that a section contains a specified percentage of students from each department or school.

Assignments may also be made using "balanced sections" as the criterion. Thus, each section would have the same number of students.

The basis for assignment may include two or more of the above criteria or even other criteria, where some flexibility is permitted in making the assignments. It is conceivable that assignments of student requests to specific sections can be programmed for the computer. We are assuming here that such assignments have already been made.

The subset of course sections, indicated by A -Identifiers, which are assigned to a student, a faculty member, or a physical plant facility, is called a request block. The order or arrangement of these blocks is immaterial. All of these request blocks are placed together on a tape file.

Example: If a faculty member(k) is to teach course sections A_1 and A_4, then his request block will be:

Faculty Member (k): A_1, A_4

Since Faculty member (k) will teach A_1 and A_4, the time schedules for A_1 and A_4 should not overlap. A_1 and A_4 must occupy disjoint time patterns.

Example: If a student (k) is assigned to sections A_1, A_5, A_6, A_7, A_{100}, then his request block will be:

Student (k): A_1, A_5, A_6, A_7, A_{100}*

Therefore, the course sections A_1, A_{100}, A_6, A_7, A_5 must have disjoint time patterns.

* Note that A_i's can be arranged in any order, i.e. it could also be A_5, A_1, A_6, A_7, A_{100}.

Example: If an auditorium (k) is to be the meeting place for the course sections A_{20}, A_{50}, A_{450}, then the request block for physical plant facility (k) will be:

Physical Plant Facility (k): A_{20}, A_{50}, A_{450}

Again, these course sections must have non-conflicting time patterns.

Flow Chart III depicts the activities of Phase II. Refer to Figure 11.

FLOW CHART III

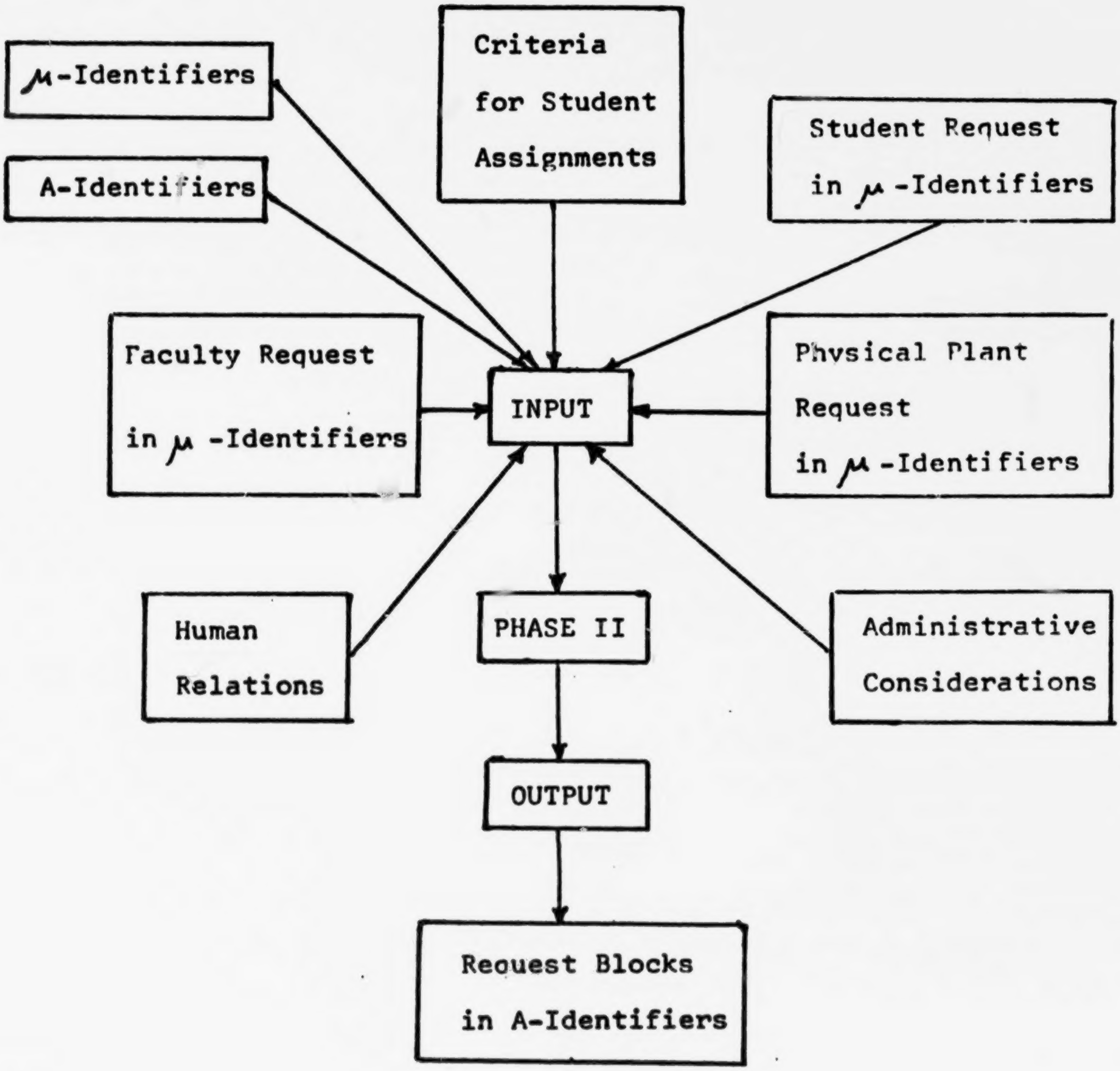

μ-Identifiers
A-Identifiers
Criteria for Student Assignments
Student Request in μ-Identifiers
Faculty Request in μ-Identifiers
INPUT
Physical Plant Request in μ-Identifiers
Human Relations
PHASE II
Administrative Considerations
OUTPUT
Request Blocks in A-Identifiers

Figure 11

PHASE III

DEVELOPMENT OF THE CONSTRAINT STRUCTURES IMPOSED ON THE SYSTEM

In this phase, a *Relations Matrix* is constructed from the request blocks developed in Phase II. The relationship of the course sections is determined with respect to the requests placed on the scheduling system by faculty, students and physical plant capabilities. From this matrix it can be determined whether a course section can be scheduled with another at the same time. The number of constraints imposed on each course section also is determined in this phase. "Number of constraints" is one of the criteria used in Phase V.

A *Relations Matrix* R, is an n x n symmetric square matrix, with the row and column headings of this matrix being the A-Identifiers of the n course sections to be scheduled. The relations matrix will be denoted symbolically by $R = [r(i,j)]$, with $i = 1,2,\ldots,n$, and $j = 1, 2, \ldots,n$, where n = total number of course sections in the system. Refer to Figure 12.

$$R = [r(i,j)] = \begin{array}{c|cccccc|} & A_1 & A_2 & \cdots & A_i & \cdots & A_n \\ \hline A_1 & 1 & r_{12} & \cdots & r_{1i} & \cdots & r_{1n} \\ A_2 & r_{21} & 1 & \cdots & r_{2i} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i & r_{i1} & r_{i2} & \cdots & 1 & \cdots & r_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_n & r_{n1} & r_{n2} & \cdots & r_{ni} & \cdots & 1 \\ \hline \end{array}$$

Figure 12

The elements of R are either 1 or 0. The diagonal of R = (1, 1,...,1). This indicates that r(i,i) = 1 for all i; i.e., a variable can have no relation constraint with itself; no constraint of the form $A_i \cdot A_i = 0$ is meaningful. The initial state of the relations matrix is R = [r(i,j) = 1] for all i and j. Thus, at the beginning of Phase III all the elements of the Relations Matrix are ones.

In this state, all the course sections could be offered together during one time unit, since there are no restrictions or conflicts indicated. But this is only the initial state.

The request blocks formed in Phase II are processed during this phase. Each request block causes the generation of C_2^p conflicts, where p equals the number of course sections in the particular request block. For example, if a student requested 5 course sections, the $C_2^5 = \frac{5!}{2!3!} = 10$, which means that 10 paired conflicts must be generated for this student.

If a request block consists of 3 course sections, A_i, A_j, A_k, then the number of paired conflicts equals $C_2^3 = 3$.

Therefore, $A_i \cdot A_j = 0$ *

$A_i \cdot A_k = 0$

$A_j \cdot A_k = 0$

*The dot (•) operator is used to indicate Boolean AND operation.

A	B	A • B
0	0	0
0	1	0
1	0	0
1	1	1

The order of each of these combinations could be changed to give:

$$A_j \cdot A_i = 0$$

$$A_k \cdot A_i = 0$$

$$A_k \cdot A_j = 0$$

In the Relations Matrix, then

$$r(i,j) = 0 \qquad r(j,i) = 0$$

$$r(i,k) = 0 \qquad r(k,i) = 0$$

$$r(j,k) = 0 \qquad r(k,j) = 0$$

As an example, suppose that an educational institution is offering 10 course sections, $A_1, A_2, \ldots, A_{10}$. (The number of sections is kept small for ease of illustration.) The starting Relations Matrix, without any constraints is as follows:

$$\begin{array}{c|cccccccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\
A_1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_{10} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Now suppose a student requests course sections A_2, A_3, A_6, A_8.

Since the time patterns for these course sections must not be overlapping, then:

$r(2,3) = 0$ $\qquad r(3,2) = 0$
$r(2,6) = 0$ $\qquad r(6,2) = 0$
$r(2,8) = 0$ $\qquad r(8,2) = 0$
$r(3,6) = 0$ $\qquad r(6,3) = 0$
$r(3,8) = 0$ $\qquad r(8,3) = 0$
$r(6,8) = 0$ $\qquad r(8,6) = 0$

With these constraints imposed on the Relations Matrix; the starting matrix of all ones is changed as shown in Figure 13.

$$\begin{array}{c|cccccccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\
\hline
A_1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
A_3 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
A_4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
A_7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_8 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
A_9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
A_{10} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Figure 13

The symmetric nature of the Relations Matrix is clear from the placement of the "zero" and "one" elements in this matrix. It was indicated previously that $r(i,i) = 1$ for all i. Furthermore, for $A_i . A_j = 0$, then both $r(i,j) = 0$ and $r(j,i) = 0$.

All the relationship constraints placed on A_i appear as zero elements in the i-th row of the Relations Matrix. If an element k of the i-th row is equal to one, $r(i,k) = 1$, then by symmetry, we conclude that the i-th element of the k-th row is equal to one, $r(k,i) = 1$. Suppose that on the basis of other criteria, it is decided that the course sections A_i and A_k should be offered during the same time period. This is possible since $r(i,k) = r(k,i) = 1$.

We would also like to know what other course sections can be scheduled during the same period with A_i and A_k. If RV is defined to be the resultant vector which contains all the constraints placed on both A_i and A_k, then

$$RV(i,k;j) = r(i;j) \cap r(k;j),$$
$$\text{for } j = 1, 2, \ldots, .n.$$

The symbol $\cap$ also denotes the logical operation "AND".

The zero members of the RV - vector indicate the course sections that cannot be scheduled when A_i and A_k are scheduled together.

For example, consider the following Relations Matrix:

$$\begin{array}{ccccc} & A_1 & A_2 & A_3 & A_4 \\ A_1 & 1 & 1 & 0 & 1 \\ A_2 & 1 & 1 & 1 & 1 \\ A_3 & 0 & 1 & 1 & 0 \\ A_4 & 1 & 1 & 0 & 1 \end{array}$$

Suppose that it is desirable (based on criteria to be discussed later) to schedule course sections A_1 and A_2 in the same time period, if possible. From the above matrix we see $r(1,2) = r(2,1) = 1$, and therefore course sections A_1 and A_2 can be

Scheduled during the same period. Now we wish to determine course sections that cannot be offered at the same time as A_1 and A_2. Thus, we determine the resultant vector for A_1 and A_2, which is:

$$\begin{array}{lcccc} & A_1 & A_2 & A_3 & A_4 \\ RV\ (1,2) = & 1 & 1 & 0 & 1 \end{array}$$

The presence of 0 for $j = A_3$ indicates that A_3 cannot be scheduled at the same time as A_1 and A_2. However, the presence of 1 in the resultant vector under $j = A_4$ denotes that course section A_4 can be scheduled in the same period with A_1 and A_2.

The row vectors of the Relations Matrix and the operation AND form an abelian semi-group. The resultant RV vector always exists such that:

1. Closure holds:

(a) $RV\ (i,k;j) = r(i;j) \cap r(k;j),$*

for $j = 1,2, \ldots, n$,

and that, (b) $RV\ (i,k,p;j) = RV\ (i,k;j) \cap r(p;j),$

for $j = 1, 2, \ldots, n$,

where $RV(i,k;j)$ on the right is the state before the AND operation, $RV\ (i,k,p;j)$ on the left refers to the state after the AND operation between $RV(i,k;j)$ and $r(p,j)$, for $j = 1,2, \ldots, n$

---

*$r(k,j)$ indicates an element of the relations matrix. $r(k;j)$ indicates the k-th row of the relations matrix.

2. Associativity holds

$$[r(i;j) \cap r(k;j)] \cap r(p;j)$$
$$= r(i;j) \cap [r(k;j) \cap r(p;j)]$$
$$= r(i;j) \cap r(k;j) \cap r(p;j)$$

for $j = 1,2,\ldots,n$. This can be extended to n vectors.

3. Commutativity holds

$$r(i;j) \cap r(k;j)$$
$$= r(k;j) \cap r(i;j)$$

for $j = 1,2, \ldots, n$.

As a result of this abelian semi-group property, it is possible to determine the course sections that can be scheduled to a particular period of the schedule cycle.

In general, we start with a course section A_i and find a course section A_k that may be offered with A_i. The resultant vector RV $(i,k;j)$ thus enables us to determine whether another course section A_p can be offered in the same period with A_i and A_k. If so, then the resultant vector RV $(i,k,p;j)$ is calculated to determine if another course section can be offered in the same period with course sections i,k, and p. This process is continued until all sections that can be offered in the same time period are exhausted from the Relations Matrix.

A Conflict Counter Vector

$$CC = \begin{bmatrix} cc_1 \\ cc_2 \\ \vdots \\ cc_i \\ \vdots \\ cc_n \end{bmatrix}$$

is appended to the Relations Matrix to indicate the number of conflicts for each course section A_i. In the i-th row for each $r(i,j) = 0$, $j = 1,2,\ldots, n$, the contents of cc_i are increased by 1. Refer to Figure 14.

$$
\begin{array}{cccccccccc}
 & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & CC \\
A_1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 4 \\
A_2 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 5 \\
A_3 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 3 \\
A_4 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 4 \\
A_5 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 3 \\
A_6 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 4 \\
A_7 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 2 \\
A_8 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 3
\end{array}
$$

Figure 14

This can be extended to provide a weighted demand for course section A_i by increasing the contents of cc_i by 1, for every conflict $A_i \cdot A_j$, $j=1,2,\ldots,n$.

It was indicated previously that the starting state for the Relations Matrix contains all ones. This indicates that no constraints are imposed on the scheduling system. All the constraints are then determined for the Relations Matrix. The constraints are generated for one row at a time.

Starting with the 1st row (say course A_1), of the initial state of the Relations Matrix, each request block from the tape file is checked to determine if the specific request block contains the course section A_1. If it does, then a zero replaces the one in the Relations Matrix for all other course sections listed in the request block. Since A_1 cannot conflict with itself, a one will still be shown for this element.

For example, if 10 course sections are to be offered by an educational institution then the starting state for the initial row (say A_1) would be:

$$\begin{array}{ccccccccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\ A_1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Suppose that the first request block from the tape file is:

$$A_1 \quad A_4 \quad A_5 \quad A_8$$

The first row could then be changed to indicate the constraints imposed by this request block to give:

$$\begin{array}{ccccccccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\ A_1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

The zeros denote that course sections A_1, A_4, A_5, A_8 cannot be offered during the same period.

The second request block is now checked to determine if it contains course section A_1. If it does, then a zero replaces the one in the Relations Matrix for all the sections contained in this request block. It is possible that a zero may have already replaced the one as a result of the constraints imposed by previous request block(s). Having already determined the first row to contain the following constraints as a result of the course sections in the first request block,

$$\begin{array}{ccccccccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\ A_1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}$$

the first row would be changed as follows if the course sections in the request block were:

$$A_1 \quad A_3 \quad A_4 \quad A_5 \quad A_7$$

$$\begin{array}{c|cccccccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} \\ A_1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

The complete tape file of request blocks is exhausted in a similar manner. If a request block does not contain a course section for the row being constructed, the complete block is ignored at this time and the next request block is checked. Thus, row 1 of the Relations Matrix has been generated to give all the constraints placed on this row. The tape file is rewound and the same procedure is utilized to develop row 2, row i, through row n. At the completion of row n, the relations matrix shows all constraints subjected on the system by the request blocks. These request blocks are for faculty and facilities, as well as for students.

The course section A-Identifiers are sorted by the numbers in the Conflict Counter vector of the Relations Matrix. The course sections are sorted so that those with the largest number of conflicts are placed first on the list.

If the course section identifiers are ordered A_s, A_t, A_v, in this new listing, then $cc_s \geq cc_t \geq cc_v$.

Thus, the list is partially ordered with respect to the number or conflicts in descending order. This new list will be called a "descending chain."

Flow Chart IV depicts the activities of this phase. Refer to Figure 15.

FLOW CHART IV

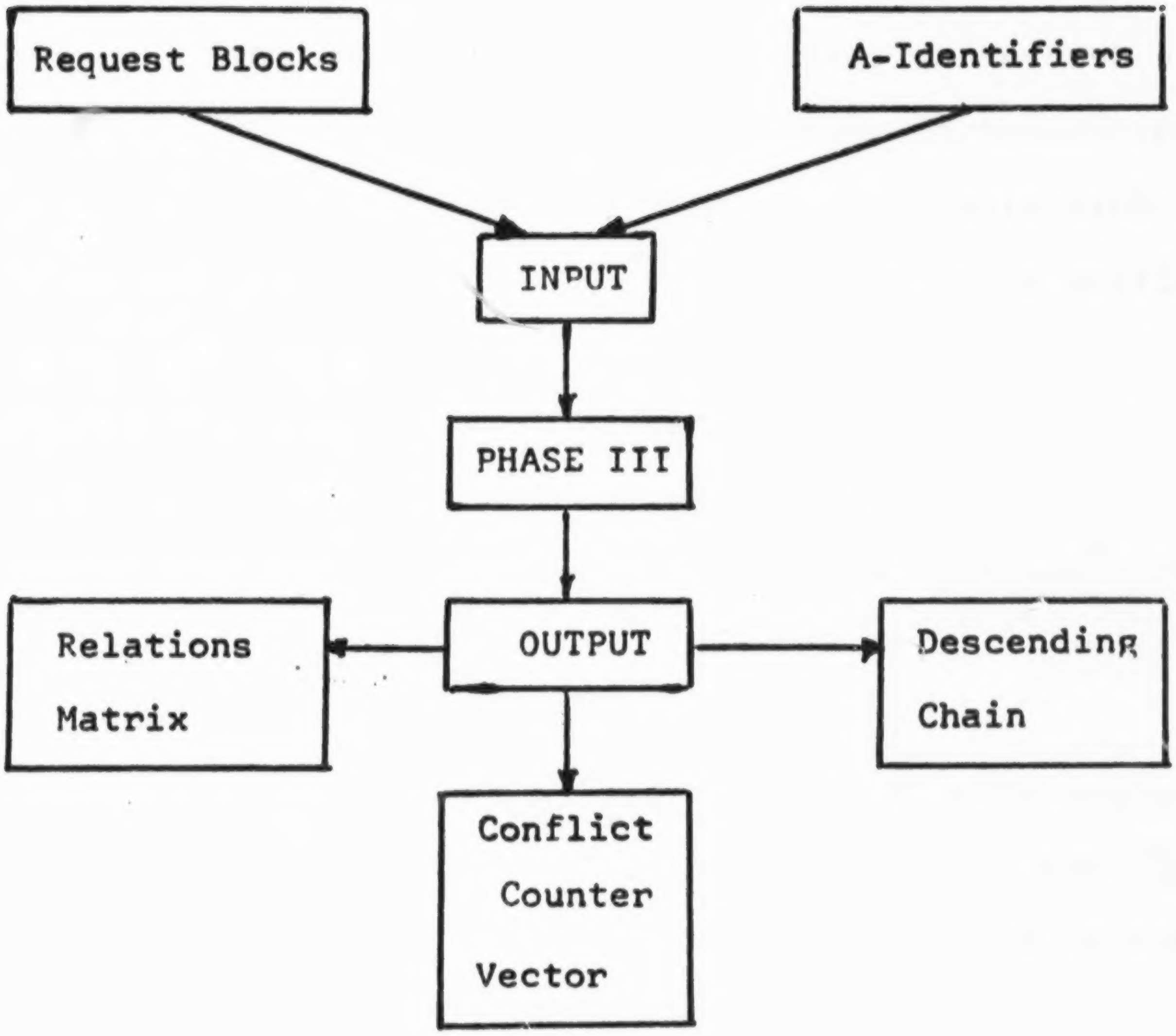

Request Blocks
A-Identifiers
INPUT
PHASE III
OUTPUT
Relations Matrix
Descending Chain
Conflict Counter Vector

Figure 15

PHASE IV

ESTABLISHMENT OF PRIORITIES BASED ON THE SYSTEM CONSTRAINTS AND THE COURSE SPECIFICATIONS

The inputs to the scheduling system in this phase are the descending chain of constraints from Phase III, and the course section specifications, which give the information needed to determine the time patterns for the courses. The course section specifications are:

Units per day
Days per cycle
Buffer day required or not

Units per day is defined to be the number of consecutive periods a course section is required to meet during a session of the day.

Since the units-per-day for a course section never changes once it is specified in the specification list, we will call the vector which gives the units per day for each course section the Units Constant vector, denoted by UC.

Thus, the A-Identifier vector for n course section has associated with it the UC vector.

A-Identifiers	UC
A_1	uc_1
A_2	uc_2
⋮	⋮
A_k	uc_k
⋮	⋮
A_n	uc_n

Each day a Units counter vector U is set with the same number of units as specified in the Units Constant vector. When one unit of a particular course section is scheduled, the unit counter associated with the particular course section is reduced by 1. This enables the computer to account for the completion of the scheduling of a course section for a certain day. Since each time a unit of the course section is scheduled the counter is reduced by 1, the course section is completely scheduled when its counter becomes zero for the day that is being worked on.

It should be noted here that only one meeting of a course section is permitted during a day.

The counter vector U is denoted as follows for the course section:

A-Identifiers	U
A_1	u_1
A_2	u_2
$\vdots$	$\vdots$
A_k	u_k
$\vdots$	$\vdots$
A_n	u_n

A meeting consists of one unit or several units that are to be scheduled within a session of a schedule day.

Days per cycle is defined to be the number of meetings a course section is to be scheduled during a schedule cycle.

A vector D indicates the number of days each course section is to meet during the schedule cycle. Thus, for the n course sections contained in the A-Identifiers vector, we have associated with it the D vector.

A-Identifiers	D
A_1	d_1
A_2	d_2
$\vdots$	$\vdots$
A_k	d_k
$\vdots$	$\vdots$
A_n	d_n

If course section A_1 is to meet three times a week, two units per meeting, then $d_1 = 3$ with $uc_1 = 2$, and $u_1 = 2$. If course section A_2 is to meet three times a week, one unit per meeting $d_2 = 3$ with $uc_2 = 1$, and $u_2 = 1$.

It may be desirable, from an academic standpoint, for some sections not to meet on consecutive days. In other words we must have at least one buffer day between meetings.

A Buffer vector, denoted by B, is constructed with 1 indicating that the course section cannot meet on consecutive days.

Thus, for the A-Identifiers of course sections we have associated the Buffer vector, where

$$b_k = \begin{cases} 0 \text{ if buffer is not required} \\ 1 \text{ if buffer is required} \end{cases}$$

A-Identifiers	B
A_1	b_1
A_2	b_2
.	.
.	.
.	.
A_k	b_k
.	.
.	.
.	.
A_n	b_n

If course section A_k had been scheduled to meet on Monday, and $b_k = 1$, then no attempt would be made to schedule the second meeting of the course section on Tuesday.

The vectors of the course section specification list are summarized in Figure 16.

A- Identifiers Course Sections	UC Units Constant	U Units Counter	D Days/ Cycle	B Buffer Day
A_1	uc_1	u_1	d_1	b_1
A_2	uc_2	u_2	d_2	b_2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
A_k	uc_k	u_k	d_k	b_k
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
A_n	uc_n	u_n	d_n	b_n

Figure 16

For example, if course section A_k meets two periods per meeting, has three meetings per cycle, with consecutive meeting days not permitted, it would be specified as:

$$uc_k = 2 \quad d_k = 3 \quad b_k = 1$$

From the Descending Chain determined in Phase III, and the D vector which indicates the number of days each course section is to meet during the schedule cycle, a priority list consisting of 5 subsets is constructed.

Subset								
1	Course	sections	which	meet	5	days	per	cycle
2	"	"	"	"	4	"	"	"
3	"	"	"	"	3	"	"	"
4	"	"	"	"	2	"	"	"
5	"	"	"	"	1	"	"	"

The five subsets of the priority list are constructed in the following way. First it should be recalled that the identifiers of the course sections are arranged by decreasing number of conflicts in the Descending Chain. The course section specification list gives the course section identifiers in the initial sequence of positive integers. Also the specification days per cycle is included in the specifications for each course section.

The procedure utilized scans the Descending Chain first and then the specification list. This scanning is repeated five different times, one for each subset.

For example: Suppose we are to construct the i-th subset.

For the k-th course section of the Descending Chain, first the k-th identifier is found in the specification list. Then a check is made to determine if $d_k = 5-i+1$; i.e., if the number of meeting days per week for the course is the same as the number specified in the subset. If $d_k=5-i+1$, then the course identifier and its specifications are moved to the i-th subset of the priority list. We repeat this check for all course sections. If $d_k \neq 5-i+1$, then we simply ignore the course section and continue immediately with the next member of the Descending Chain.

After all course sections have been slotted into one of the 5 subsets by this procedure, the course sections within each subset are already in descending order, with the course section in the subset having the greatest number of conflicts at the top of the order, and the course section with the least number of conflicts at the bottom.

In a sense we form five partially ordered Descending Chains with respect to the Descending Chain obtained in Phase III.

We know from the generation of the subsets that:

(1) All course sections which meet 5 days per cycle are in Subset 1; all course sections which meet 4 days per cycle are in Subset 2;...; all course sections which meet one day per cycle are in Subset 5.

(2) In the i-th subset, the number of conflicts for course section $A_k \geq$ the number of course sections for A_{k+1}.

(3) The number of conflicts for the k-th course section in subset i may be $<$, $=$, $>$, the number

of conflicts for the k-th course section of any subset $\neq i$.

This arrangement of subsets and course sections within the subsets is called the Priority List.

Flow Chart V depicts the activities of Phase IV. Refer to Figure 17.

FLOW CHART V

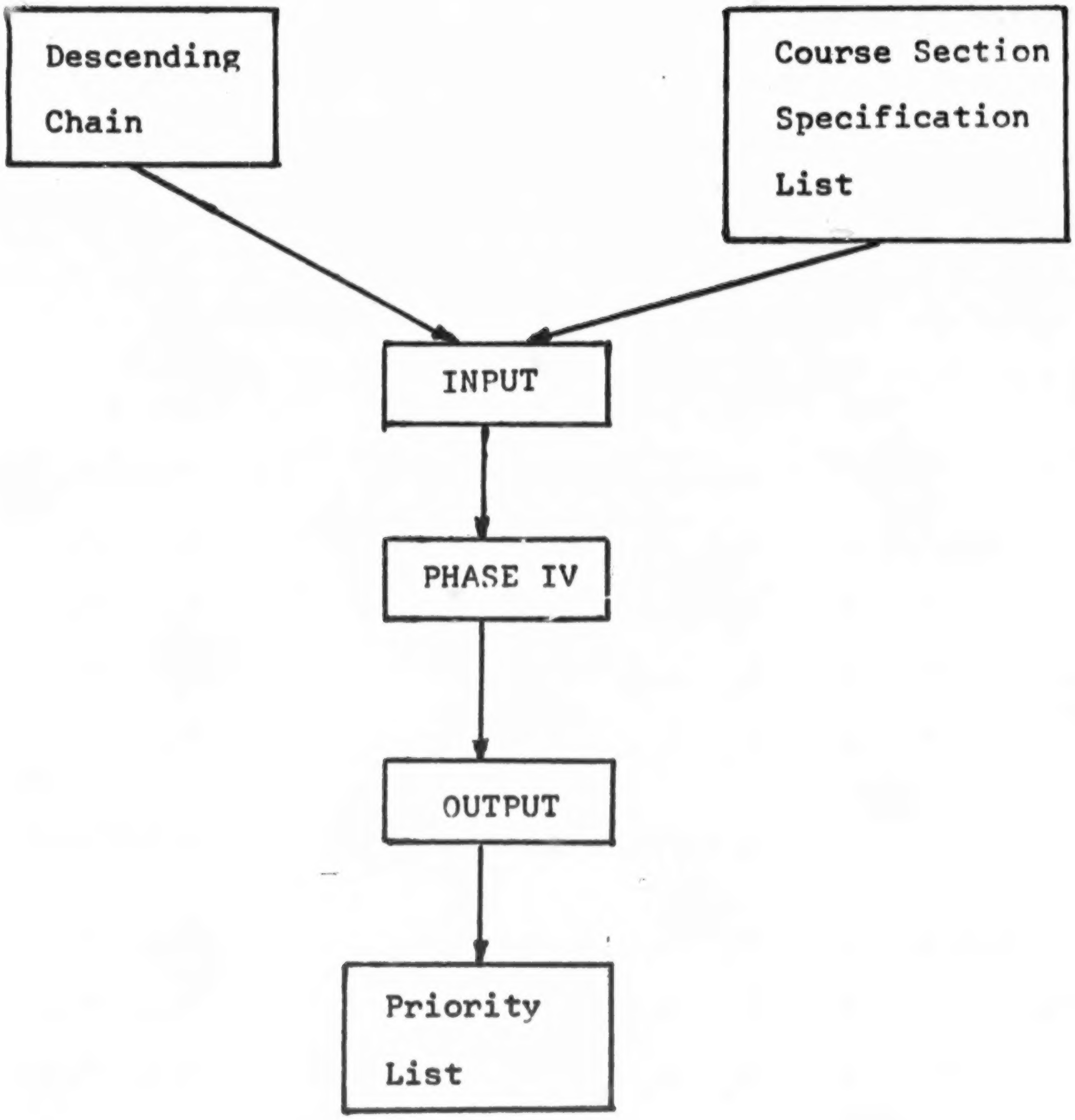
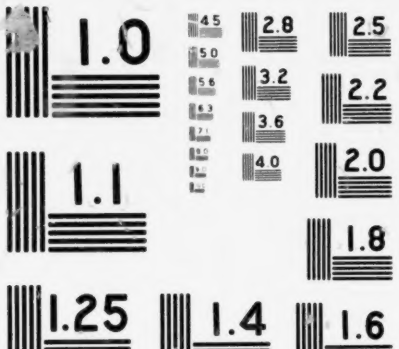

Descending Chain
Course Section Specification List
INPUT
PHASE IV
OUTPUT
Priority List

Figure 17

3 OF 7

ED001255


1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963

PHASE V

CONSTRUCTION OF THE MASTER SCHEDULE TO SATISFY THE CONDITIONS ESTABLISHED IN PHASE I THROUGH IV.

Since the Relations Matrix and the Priority List have been determined, it is now possible to construct the Master Schedule.

The Master Schedule gives for each period of each schedule day, the course sections which are to meet at that time. Figure 18 is an example of a Master Schedule for two days, with the course sections specified by their A-Identifiers.

		Day 1	Day 2
		Morning Session	Morning Session
Period	1	A_6 A_{15} A_{19}	A_6 A_1
"	2	A_8 A_{22}	A_1 A_{22}
"	3	A_8 A_{13}	A_1 A_3
"	4	A_8 A_{13}	A_3
		Afternoon Session	Afternoon Session
Period	5	A_7 A_4 A_{12}	A_{16} A_{10}
"	6	A_4 A_{10} A_{16}	A_{13} A_{15}
"	7	A_4 A_5	A_{15} A_{14}
"	8	A_4 A_5	A_{14} A_9 A_{11}

Figure 18

Provision is made in the program to accommodate two sessions per day, such as morning and afternoon. The meeting of a course section will be assigned to only one of the sessions for a particular day. Thus, if a course section meeting is three units per day, then the course section will be assigned three consective periods in either the morning or in the afternoon sessions.

Since the number of periods for the sessions vary from one educational institution to another, the number of periods assigned to the morning session and the number assigned to the afternoon session, must be specified as parameters to the system.

A counter is associated with each one of these parameters. The counter J is associated with the number of periods in the morning session, and the counter J' is associated with the number of periods in the afternoon session.

The counter I is associated with the number of periods within a schedule day. Counter I is the sum of the periods in the morning and afternoon sessions, i.e., I =J+J'. At the beginning of each schedule day, the J counter is loaded with the number of periods in the morning session, and the I counter is loaded with the number of periods in both the morning and afternoon sessions. At the start of the afternoon session, the J'counter is loaded with the number of periods in the afternoon session.

The I and J counters are decreased by one at the end of each period in the morning session, with the I and J' counters decreased by one at the end of each period in the afternoon session. This periodic decrease enables the computer to keep track of the number of periods remaining in the schedule day and in each session of the schedule day. If, for instance, I = 9, and J = 4 at the beginning of the morning session, then I = 7 and J = 2 will be indicated on the counters at the end of the second period of the day. When J = 0, I will be equal to 5, at which time J' will be loaded with 5. At the end of the afternoon session, J' = 0 and I = 0 will be shown on the counters.

Some educational institutions desire the meeting of classes at noon. This special case is covered. It is assumed that the morning session is equal to the schedule day and there are no periods in the afternoon session. When the parameters are specified, the number of periods within a schedule day are assigned to the morning session and no period is assigned to the afternoon session.

During the construction phase, it must be possible to indicate the completion of the various stages in the development of the Master Schedule. It must be known, as the Master Schedule is being constructed, which course sections have been scheduled and whether they are partially scheduled or completely scheduled. Some may have been scheduled for the cycle, some only for a day, and still others only for a period of a day.

If a course section meets only one period per cycle. then this section is deleted from the scheduling process as soon as one unit of this course section has been assigned to a period.

A vector denoted by WV is utilized to indicate whether a course section has been completely scheduled for the cycle or not. This vector has a one-to-one correspondence with the course sections.

The WV vector is associated with the n course sections, as follows:

A -Identifiers	WV
A_1	wv_1
A_2	wv_2
.	.
.	.
.	.
A_k	wv_k
.	.
.	.
.	.
A_n	wv_n

Initially the WV vector contains all ones. WV= (wv_k =1), for k =1,2,...,n. This means that no course section has been scheduled for the cycle. If a course section A_k has been completely scheduled for the cycle, then wv_k =0.

When the WV vector consists of all zeros, the scheduling process ends since all the course sections have been completely scheduled for the cycle.

It is also important to know the day(s) of the week a course section has been scheduled. If the meeting of a course section has been scheduled for a particular day, scheduling of additional meetings should not be considered for that day.

A vector denoted by DV is used to determine whether a meeting of the course section has been scheduled for the particular day or not. This vector is also in one-to-one correspondence with the course sections shown as follows:

A-Identifiers	DV
A_1	dv_1
A_2	dv_2
$\vdots$	$\vdots$
A_k	dv_k
$\vdots$	$\vdots$
A_n	dv_n

At the beginning of each new day the DV vector is initialized to contain all ones,

$$DV = (dv_k = 1), \text{ for } k = 1, 2, \ldots, n.$$

This means that all course sections may be scheduled for the day. All course sections are assigned a one, even though one or more course sections may have been completely scheduled prior to this day. If course section A_k had been completely scheduled before this day, then in the week vector WV, $wv_k = 0$. At the start of each period for the new day, the logical AND operation is performed on WV and DV. Consequently, if wv_k=0, and dv_k=1, the element in the resultant vector equals zero, which indicates the course section A_k is not to be scheduled in this period. This would be true for all periods of the day, with the result being that the course section A_k could not be scheduled during this day. When a meeting for a course section has been scheduled on that day, then $dv_k = 0$, which indicates that a complete meeting of course section A_k has been scheduled for that day, and cannot be scheduled in any following periods of this particular day.

In Phase IV it was indicated that some course sections may not be permitted to have meetings scheduled on successive days. This was denoted by a Buffer vector, where, for A_k, the k-th element in this Buffer vector equals one, if buffer day is required, and zero if no buffer day is needed for the course section. Thus, if $b_k = 0$, then course section A_k is permitted to be scheduled that day; however, if $b_k = 1$, then A_k cannot be scheduled that day since it is a buffer day for this course section.

Two additional vectors, BC1 (Buffer Check 1) and BC2 (Buffer Check 2) are used so that the buffer day requirement is satisfied.

A-Identifier	BC1	BC2
A_1	$bc1_1$	$bc2_1$
A_2	$bc1_2$	$bc2_2$
.	.	.
.	.	.
.	.	.
A_k	$bc1_k$	$bc2_k$
.	.	.
.	.	.
.	.	.
A_n	$bc1_n$	$bc2_n$

At the start of the scheduling cycle both BC1 and BC2 are initialized with ones:

$BC1 = (bc1_k = 1)$, for $k = 1,2,\ldots, n$, and
$BC2 = (bc2_k = 1)$, for $k = 1,2,\ldots, n$.

This means that on the first day, any course section can be scheduled and not violate the buffer constant.

When a course section A_k, for which $b_k = 1$ (indicating a buffer day is required for this course section), satisfies all other restrictions, a check is made to determine if $bc2_k = 1$ or if $bc2_k = 0$. If $bc2_k = 1$, then $bc1_k$ is changed from 1 to 0, and A_k is scheduled. When a course section A_k, for which $b_k = 0$, satisfies all other restrictions, no check is made on $bc2_k$, and $bc1_k$ is not changed (i.e.; $bc1_k$ is left equal to one) and A_k is scheduled.

In vector BC1, at the end of any schedule day. $bc1_k = 1$ indicates A_k can be scheduled following day, and not violate buffer constraint, and $bc1_k = 0$ indicates A_k cannot be scheduled the following day, since otherwise the buffer constraint would be violated. At the end of any schedule day, the elements of vector BC2 are replaced by the contents of vector BC1, and the vector BC1 is again initialized with ones. This transfer carries the buffer constraint conditions from one schedule day to the next. The convention established for interpretation of 0 and 1 in previous vectors still holds. If, for a particular day, $bc2_k = 1$, the k-th course section is permitted to be scheduled on that day, whereas, if $bc2_k = 0$, then the k-th course section cannot be scheduled on that day.

For a course section which has more than one unit to be assigned per meeting (when it is necessary to have consecutive periods), it is important to know if enough remaining periods for a particular session are available to schedule this multi-period meeting. For example, suppose the schedule for the third period of the morning session of a schedule day is being determined, and the parameter J = 4, (which means there are a total of four periods in the morning session). In addition consider course section A_k which requires three consecutive periods per meeting. As a result, since only two periods remain in the morning session, this course section A_k cannot be scheduled in either the third or fourth periods of the morning session.

To indicate whether or not sufficient periods are available to schedule a complete meeting of the course section in a particular session, an AV vector is established. The elements of AV are in one-to-one correspondence with the A-Identifiers. The association between the A-Identifiers and the AV vector follows:

A-Identifiers	AV
A_1	av_1
A_2	av_2
$\vdots$	$\vdots$
A_k	av_k
$\vdots$	$\vdots$
A_n	av_n

At the beginning of each morning session, the AV vector is initialized with all ones,

$$AV = (av_k = 1), \text{ for } k = 1,2,\ldots, n.$$

This means that sufficient periods are available to schedule a complete meeting for all course sections. All course sections are assigned a one even though one or more course sections may have been scheduled prior to this day, or may have been scheduled prior to the period being scheduled for the particular day. A logical AND operation is performed between WV, DV, and AV to determine course sections which can be scheduled in a particular period.

It should not be interpreted that all course sections can be scheduled in this session, since other constraints must also be satisfied before a course section is scheduled. If, in the scheduling for any period of a particular session, an insufficient number of periods remain to schedule course section A_k, then $AV_k = 0$. It should be noted that the morning and afternoon sessions are disjoint sets.

Whenever a unit of a course section which meets two or more consecutive periods has been scheduled, the scheduling program must ensure that the remaining units of this meeting for the course section are scheduled in the following consecutive periods. Suppose that a course section, which meets three consecutive periods per meeting, has its first unit scheduled in the second period of the morning session. We must be sure that the scheduling procedure will schedule the next two units of this course section in the third and fourth periods of the morning session, since the units per meeting must be scheduled consecutively.

Two check vectors, CV1 and CV2 are designed to facilitate the checking of assignments to course sections which require more than one period per meeting. The elements of CV1 and CV2 are in one-to-one correspondence with the course sections.

A-Identifiers	CV1	CV2
A_1	$cv1_1$	$cv2_1$
A_2	$cv1_2$	$cv2_2$
$\vdots$	$\vdots$	$\vdots$
A_k	$cv1_k$	$cv2_k$
$\vdots$	$\vdots$	$\vdots$
A_n	$cv1_n$	$cv2_n$

At the beginning of each morning and afternoon session, these vectors are initialized with zeros,

$CV1 = (cv1_k = 0)$, for $k = 1,2,\ldots, n$

$CV2 = (cv2_k = 0)$, for $k = 1,2,\ldots, n$.

If the A_k is a course section which requires consecutive periods per meeting, then $cv1_k = 1$ when its first unit is scheduled. Before the scheduling of the next period begins, the CV2 vector is made equal to the CV1 vector, and CV1 is initialized with zeros.

These two check vectors are necessary since the following two operations are performed concurrently:

(1) If the k-th element in CV2 equals one, then course section A_k, and all other course sections which are designated by a one in CV2, must be scheduled in this period before any other course sections.

(2) When the course section A_k is being scheduled in this period, and if another unit of A_k must be scheduled in the period to follow this one, then $cv1_k$ is changed from zero to one. Again, before the scheduling of the following periods begins, CV2 vector is made equal to CV1 vector, CV1 is initialized with zeros and the cycle is repeated.

A vector denoted by RV is used to determine that either a unit of a course section may be scheduled for the particular period of a day or it cannot be scheduled. This vector is also in one-to-one correspondence with the course sections shown as follows:

A-Identifiers	RV
A_1	rv_1
A_2	rv_2
⋮	⋮
A_k	rv_k
⋮	⋮
A_n	rv_n

At the beginning of each new period the RV vector is initialized to contain all ones,

$$RV = (rv_k = 1), \quad \text{for } k = 1,2,\ldots, n.$$

This means that all course sections may be scheduled to the period. The previous scheduling decisions are carried into effect when the logical "AND" operations are performed between the RV vector and the other vectors considered in this phase. The RV' vector is used to temporarily designate the conditions of the RV Vector after the logical operations. Immediately after a logical operation, RV vector is made equal to RV' vector, and hence readied for the next logical operation.

Until now this final phase of the program has been a construction phase, since all vectors have been prepared and arranged to facilitate the construction of the Master Schedule on the computer. The master schedule generation is an iterative program, starting with the first period of the first schedule day. The iterative procedure is repeated for each successive period until all course sections have been scheduled, without violating any constraints of the scheduling system. The number of periods so obtained to schedule all course sections is minimal for the assignment policy used for multi-section courses.

Flow Chart VI depicts the activities of this phase. Refer to Figure 19.

FLOW CHART VI

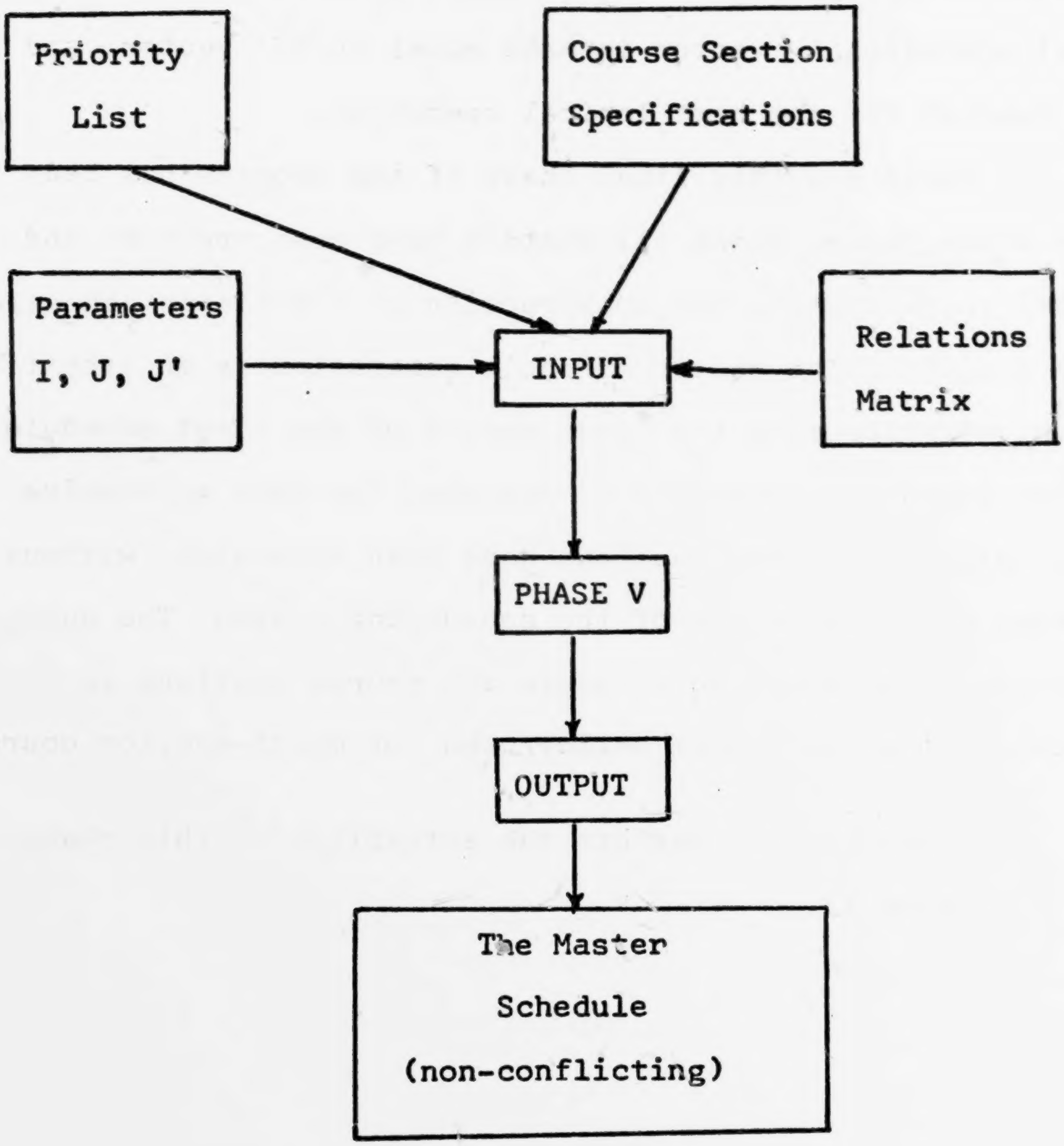

Priority List
Course Section Specifications
Parameters I, J, J'
INPUT
Relations Matrix
PHASE V
OUTPUT
The Master Schedule (non-conflicting)

Figure 19

ITERATION PROCEDURE

The general iteration procedure for the computer to generate the Master Schedule will now be developed. It is desirable at this time to give a composite tabulation of the initial status of the scheduling structure developed previously for the Master Schedule.

(1) The Relations Matrix,

$$R = [r(i,j)], \text{ for } i = 1,2,\ldots, n, \text{ and } j = 1,2,\ldots, n,$$

has been determined. The matrix is not altered in any way throughout the execution of the entire scheduling process.

(2) The Course Specification List has been developed:

A- Identifiers Course Sect.	UC Units Constant	U Units Counter	D Days/ Week	B Buffer Day
A_1	uc_1	u_1	d_1	b_1
A_2	uc_2	u_2	d_2	b_2
⋮	⋮	⋮	⋮	⋮
A_k	uc_k	u_k	d_k	b_k
⋮	⋮	⋮	⋮	⋮
A_n	uc_n	u_n	d_n	b_n

(3) Course sections have been arranged in a Priority List

Subset	
1	Course sections which meet 5 days per cycle
2	" " " " 4 " " "
3	" " " " 3 " " "
4	" " " " 2 " " "
5	" " " " 1 day " "

All course sections in a subset are ordered by number of conflicts obtained from the Descending Chain. The course section with the greatest number of conflicts has the highest priority for that particular subset, and the one with the least number of conflicts has the lowest priority.

(4) Provision has been made in the scheduling system to update the variables as required in the iterative procedure. The following vectors have been constructed to facilitate the monitoring of the scheduling function.

a. WV vector indicates whether or not a course section has been completely scheduled for the cycle.

$$wv_k = \begin{cases} 1 & \text{if k-th course section has not been scheduled for the cycle} \\ 0 & \text{if k-th course section has been scheduled for the cycle} \end{cases}$$

b. DV vector shows whether or not a complete meeting of a course section has been scheduled for the particular day being scheduled.

$$dv_k = \begin{cases} 1 & \text{if k-th course section does not have complete meeting scheduled for the particular day} \\ 0 & \text{if k-th course action does have complete meeting scheduled for the particular day} \end{cases}$$

c. AV vector indicates whether or not sufficient periods are available in the session (morning-afternoon) to schedule a complete meeting of a course section.

$$av_k = \begin{cases} 1 & \text{if sufficient number of periods remain in the session to schedule a complete meeting of the k-th course section} \\ 0 & \text{if not sufficient number of periods remain in the session to schedule a complete meeting of the k-th course section} \end{cases}$$

d. CV1 vector shows the course sections which must be scheduled in the next period before any other course sections.

$$cv1_k = \begin{cases} 1 & \text{if course section must be scheduled in next period} \\ 0 & \text{if course section is not required to be scheduled in next period} \end{cases}$$

e. BC1 vector exhibits whether or not A_k can be scheduled the following day and not violate the buffer constraints.

$$bv1_k = \begin{cases} 1 & \text{if the k-th course section can be scheduled the following day and not violate buffer constraint} \\ 0 & \text{if the k-th course section can not be scheduled the following day} \end{cases}$$

f. RV vector exibits that either A_k may be scheduled to a period of a schedule day or it cannot be scheduled.

$$rv_k = \begin{cases} 1 \text{ if the k-th course section may be scheduled to the period} \\ 0 \text{ if the k-th course section cannot be scheduled to the period} \end{cases}$$

Following is the initial status of the vectors in the scheduling system; i.e., at the start of the first period of the first day:

$wv_k=1$, for $k=1,2,\ldots,n$

$dv_k=1$, "

$av_k=1$, "

$cv1_k=0$, "

$cv2_k=0$, "

$bv1_k=1$, "

$bv2_k=1$, "

$rv_k=1$, "

When the system is in the starting state, a unit of the first course section, which is at the top of the subset of the priority list, will be scheduled to the first period of the first day of the schedule cycle. Scheduling of other course sections during this period will be effected by the row vector of this first course section in the Relations Matrix. Each time a course section is scheduled the state of the constraints is tightened by the additional constraints of the newly scheduled course section.

It should be emphasized that once we start at the beginning of the priority list we schedule all the course sections in subset 1 which can be scheduled into this period. Then we proceed to schedule all the course sections that can be scheduled into this period from subset 2, subset 3, subset 4, and subset 5, respectively. Further, it is pointed out that a unit of a course section that may require 5, or 4, or 3, or 2, or 1 schedule days may be scheduled with any other course section which has different specifications, provided all restrictions imposed on the system are satisfied. A unit of a course section that requires one unit per day and that of another which requires consecutive units per day may also be scheduled in the same period of the cycle.

The first iteration is terminated when no more course sections can be scheduled to the first period. At the end of the first iteration the RV vector (the resultant vector) will have a 1 for each course section for which a unit has been scheduled in this period; and it will have a 0 for each course section not scheduled in this period. If a unit has been scheduled for a course section which requires more than one consecutive period per meeting, then it and all other such course sections are scheduled for the next period before any other courses. In other words, these course sections are forced to be scheduled in the next period. Since these course sections have satisfied the constraints when scheduling their first unit, the next consecutive unit of each can also be assigned without violating the constraints of the system.

An overview of the scheduling procedure has been presented above. Now a detailed explanation of the program and computer bookeeping required for an iteration will be developed. Each scheduling period requires an iteration, which consists of four possible functions.

Function 1: Selection of first course section to be scheduled in period i if no course sections are forced. This operation is used only if no courses are partially scheduled in period i-1.

Function 2: Selection of remaining course sections to be scheduled in period i from among course sections not forced.

Function 3: Selection of first course to be scheduled in period i if one or more course sections are forced. A forced course section has a consecutive period meeting which has been partially scheduled in period i-1.

Function 4: Selection of remaining course sections to be scheduled in period i from among course sections forced.

It should be recalled that at the start of each iteration the RV vector is initialized to contain all ones.

$$RV=(rv_k=1) \quad , \text{ for } k=1,2,\ldots,n.$$

The first step in scheduling course sections to period i is to determine if at least one course section is partially scheduled. Has the previous iteration (period i-1) scheduled a unit of a course section which forces another unit of this course section to be scheduled in period i? Does the CV2 vector contain at least one non-zero element? If yes, go to Function 3; if no, go to Function 1. This is a branching operation which indicates the procedure to use to schedule the first course section in period i.

Function 1

a. Start scanning the priority list from the top to find a course section A_k which satisfies the following criteria:

(1) $wv_k = 1$. This indicates the course section A_k has not been scheduled for the cycle.

(2) $dv_k = 1$. This indicates A_k has not been scheduled for the day.

(3) $av_k = 1$. This indicates that it may be possible to schedule A_k during this session. Another check as outlined in (b) is necessarv to determine if sufficient periods remain in the sessions to schedule the complete meeting of A_k*.

If the k-th element in any one of these vectors is zero, then this course section cannot be scheduled in the i-th period. We continue to scan the priority list until a course section A_k is found which satisfies these criteria.

---

* Consider the example where the morning session contains three periods and the afternoon session contains four periods, and we are presently scheduling course sections for the first meeting of the morning session. If a meeting of A_k required four periods, av_k=1 since the AV vector has been iniatilized with ones at the start of the session. However, A_k could not be scheduled in this session because there are only three periods in the morning session. Thus, the fact that av_k=1 does not insure that sufficient periods are available to schedule a complete meeting of a course section requiring more than one consecutive period.

b. Determine if sufficient periods are available in the session to schedule A_k.

Is $u_k \leq$ J(or J')?, where u_k is the k-th element of the unit counter vector, U, indicating the number of units of A_k remaining to be scheduled for a complete meeting, and J (or J') is the number of periods in the morning session, or afternoon session, whichever is applicable.

If $u_k >$ J (or J'), then A_k cannot be scheduled in period i or any other subsequent period during this session. av_k=1 is changed to av_k=0, and we start again with step (a).

If $u_k \leq$ J(or J') continue with step (c).

c. Determine if course section A_k requires a buffer day between its meetings.

If b_k = 1, this indicates a buffer day is required. A check must be made to determine if A_k has been scheduled the previous day. If, in the buffer check vector BC2, $bc2_k$ = 1, then a meeting of course section A_k can be scheduled in the day which contains the i-th period. In the buffer check vector BC1, make $bc1_k$=0, so that this course section will not be scheduled the next day. Then proceed with step (d).

If $bc2_k$=0, then A_k cannot be scheduled since the day which contains the i-th period must be a buffer day. In the vector RV, make rv_k=0 to indicate that A_k cannot be scheduled. Start again with step (a).

If $b_k=0$, then A_k does not require a buffer day between its meetings, and we proceed with step (d).

d. Enter the course section A_k in the Master Schedule for this period.

e. (1) From the Relations Matrix obtain the row vector for A_k, denoted by $r(k;j)$, for $j = 1,2,\ldots, n$.

(2) Determine the other course sections which may possibly be scheduled in the same period with A_k by obtaining the resultant vector RV'*:

$RV'(j) = r(k;j) \cap WV(j) \cap DV(j) \cap AV(j) \cap RV(j)$, for $j = 1,2, \ldots, n$.

It is well to recall that the priority list of course sections was prepared with respect to the number of days per week that each section meets, and the number of conlficts as contained in the Relations Matrix for each course section. Then, the courses which meet 5 days per week are in Subset 1; those which meet 4 days per week are in Subset 2; 3 days per week in Subset 3; 2 days per week in Subset 4; and one day per week in Subset 5. Within each subset, the course sections are ordered by number of conflicts, with the course section having the largest number of conflicts placed at the top of the list for each subset. The first course section A_k to be scheduled in period i is the course section which meets the maximum number of days in a cycle, and which has the maximum number of conflicts in the relations matrix. A_k is among those course sections which lie in the intersection

---

* It should be recalled that the vector RV always equals the vector RV' which logically precedes RV in the iteration. In an equation the vector RV' is the resultant of the vector RV and other vectors.

$$WV(j) \cap DV(j) \cap AV(j) \cap RV(j),$$
$$\text{for } j = 1,2,\ldots,n.$$

The criterion of optimality is dictated by the priority list ordering of the course sections. The scheduling of A_k usually restricts the subset $WV(j) \cap DV(j) \cap AV(j) \cap RV(j)$ by $r(k;j)$.

f. Decrease the units per day by one since a unit of A_k has been scheduled for the i-th period.

$$u_k = u_k - 1$$

If $u_k=0$, then all units of course section A_k have been scheduled for the day, and we proceed with step (g).

If $u_k > 0$, then another unit of course section A_k must be scheduled in the next period. Thus, cvl_k is set equal to 1, and we proceed with Function 2.

g. Indicate that A_k has been scheduled for the day by setting $dv_k=0$.

h. Decrease the days per cycle by one since A_k has been scheduled for the day containing the i-th period.

$$d_k = d_k - 1.$$

If $d_k = 0$, then A_k has been scheduled for the cycle, wv_k is set equal to zero, and we go to Function 2.

If $d_k > 0$, then the course section A_k has not been scheduled for the cycle. Replace the units per day counter with the units per day constant.

$u_k = uc_k$

Proceed with Function 2.

Function 2

In this function the remaining course sections to be scheduled in period i are selected from among course sections not forced.

a. Continue scanning the priority list from the top to find a new course section $A_{k'}$ which can be scheduled in the i-th period.

   If $cv2_{k'} = 0$, this means that $A_{k'}$ was not forced and we proceed with step (b).

   If $cv2_{k'} = 1$, this means that $A_{k'}$ was forced and has already been scheduled by Function 3 or by Function 4. Continue scanning the priority list.

b. Determine if the resultant vector RV contains a 1 or 0 for the k' element in this vector.

   If $rv_{k'} = 0$, the $A_{k'}$ cannot be scheduled for the i-th period since it would violate at least one of the constraints. Go to step (a).

   If $rv_{k'} = 1$, then it may be possible for course $A_{k'}$ to be scheduled in period i. Proceed with step (c).

c. Determine if sufficient periods are available in the session to schedule $A_{k'}$.

   Is $u_{k'} \leq J$ (or J')?, where $u_{k'}$ is the k-th element of the unit counter vector, U, indicating the number of units of $A_{k'}$ remaining to be scheduled for a complete meeting and J (or J') is the number of periods in the morning session or the afternoon session, whichever is applicable. If $u_{k'} > J$(or J'), then $A_{k'}$ cannot be scheduled in period i or any other subsequent period during the session. Then $av_{k'} = 1$ is changed to $av_{k'} = 0$, and we start again with step (a).

If $u_{k'} \leq J$(or J') continue with step (d).

d. Determine if course section $A_{k'}$ requires a buffer day between its meetings.

If $b_{k'} = 1$, this indicates a buffer day is required. A check must be made to determine if $A_{k'}$ has been scheduled the previous day. If in this buffer check vector BC2, $bc2_{k'} = 1$, then a meeting of course section $A_{k'}$ can be scheduled for the day which contains the i-th period. In the buffer check vector BC1, make $bc1_{k'} = 0$, so that this course section will not be scheduled the next day. Then proceed with step (e).

If $bc2_{k'} = 0$, then $A_{k'}$ cannot be scheduled, since the day which contains the i-th period must be a buffer day. In the vector RV, make $rv_{k'} = 0$ to indicate that $A_{k'}$ cannot be scheduled. Start again with step (a).

If $b_{k'} = 0$, then $A_{k'}$ does not require a buffer day between its meetings, and we proceed with step (e).

e. Enter the course section $A_{k'}$ in the Master Schedule for this period.

f. Determine the other course sections which may be scheduled with $A_{k'}$ and the previously scheduled section(s), by obtaining a new resultant vector,

$$RV'(j) = RV(j) \cap r(k';j),$$

for $j = 1,2,\ldots, n$.

It should be noted that each course section $A_{k'}$ scheduled now is the course section which has maximum priority. $A_{k'}$ is among those course sections which lie in the intersection $RV(j) \cap \overline{CV2(j)}$, for $j = 1,2,\ldots, n$. The criterion of optimality is again dictated by the priority list ordering of the course sections. The scheduling of each course section $A_{k'}$ usually restricts the subset $RV(j) \cap \overline{CV2(j)}$ to a new subset $RV(j) \cap \overline{CV2(j)} \cap r(k';j)$, for $j = 1,2,\ldots, n$.

g. Decrease the units per day by one since a unit of $A_{k'}$ has been scheduled for the i-th period.

$$u_{k'} = u_{k'} - 1$$

If $u_{k'} = 0$, then all units of course section $A_{k'}$ have been scheduled for the day, and we proceed with step (h).

If $u_{k'} > 0$, then another unit of course section $A_{k'}$ must be scheduled in the next period. Thus, $cv1_{k'}$ is set equal to one, and we proceed with step (a).

h. Indicate that $A_{k'}$ has been scheduled for the day by setting $dv_{k'} = 0$.

i. Decrease the days per cycle by one, since $A_{k'}$ has been scheduled for the day containing the i-th period.

$$d_{k'} = d_{k'} - 1$$

If $d_{k'} = 0$, then $A_{k'}$ has been scheduled for the cycle.

Then $wv_{k'}$ is set equal to zero, and we proceed with step (a).

If $d_{k'} > 0$, then the course section $A_{k'}$ has not been scheduled for the cycle. Replace the units per day counter with the units per day constant.

$$u_{k'} = uc_{k'}$$

Proceed with step (a). Function 2 is repeated until the priority listing of course sections is exhausted. The final resultant vector will have a one for each course section scheduled in the i-th period, and a zero for each course section not scheduled in the i-th period.

When the iteration for the i-th period is completed, we have obtained the A-Identifiers of course sections to be scheduled in the i-th period. All vectors are initialized as described previously, and the iteration for period i+1 begins.

Function 3

It was stated earlier that the first step in scheduling course sections was to determine if at least one course was partially scheduled in period i-1. Thus, if at least one element of the CV2 vector contains a one, we start with Function 3. We know that every course section which has a one in the CV2 vector is forced to be scheduled in this period. Function 3 schedules only the first of these forced courses. Function 4 schedules the remaining forced courses. We now consider Function 3.

a. Starting at the top of the priority list, determine the first course section A_k for which $cv2_k = 1$.

b. Enter the course section A_k in the Master Schedule for this period.

c. From the Relations Matrix obtain the row vector for A_k, denoted by

$$r(k;j), \quad \text{for } j = 1,2,\ldots, n.$$

Determine the other course sections which may possibly be scheduled in the same period with A_k by obtaining the resultant vector:

$$RV'(j) = r(k;j) \cap WV(j) \cap DV(j) \cap AV(j) \cap RV(j)$$

for $j = 1,2,\ldots, n$.

d. Decrease the units per day by one since a unit of A_k has been scheduled for the i-th period.

$$u_k = u_k - 1$$

If $u_k = 0$, then all units of course section A_k have been scheduled for the day, and we proceed with step (e).

If $u_k > 0$, then another unit of course section A_k must be scheduled in the next period. Thus, $cv1_k$ is set equal to 1, and we proceed with Function 4.

e. Indicate that A_k has been scheduled for the day by setting $dv_k = 0$.

f. Decrease the days per cycle by one since A_k has been scheduled for the day containing the i-th period.

$$d_k = d_k - 1$$

If $d_k = 0$, then A_k has been scheduled for the cycle. Then wv_k is set equal to zero, and we go to Function 4.

If $d_k > 0$, then the course section A_k has not been scheduled for the cycle. Replace the units per day counter with the units per day constant.

$$u_k = uc_k$$

Proceed with Function 4.

Function 4

In Function 3 only the first A_k which has $cv2_k = 1$, is processed. However, all other course sections which have a one in the CV2 vector are also forced, and must be scheduled in the same period. The course sections which are forced have been scheduled in the i-1 period, and the necessary checks have already been made to insure that no constraints are violated when scheduling these courses in the same period. These course sections are scheduled in the following manner.

a. Continue to scan the priority listing from the top until we find a new $A_{k'}$ for which $cv2_{k'} = 1$. Place $A_{k'}$ in the Master Schedule for this period.

b. Determine the other course sections which may be scheduled with $A_{k'}$ and the previously scheduled course(s) A_k by obtaining a new resultant vector.

$$RV'(j) = RV(j) \cap r(k';j),$$

for $j = 1,2,\ldots, n$.

c. Decrease the units per day by one since a unit of $A_{k'}$ has been scheduled for the i-th period.

$$u_{k'} = u_{k'} - 1$$

If $u_{k'} = 0$, then all units of course section $A_{k'}$ have been scheduled for the day and we proceed with step (d).

If $u_{k'} > 0$, then another unit of course section $A_{k'}$ must be scheduled in the next period. Thus, cvl_k is set equal to one, and we proceed with step (a).

d. Indicate that $A_{k'}$ has been scheduled for this day by setting $dv_{k'} = 0$.

e. Decrease the days per cycle by one since $A_{k'}$ has been scheduled for the day containing the i-th period.

$$d_{k'} = d_{k'} - 1$$

If $d_{k'} = 0$, then $A_{k'}$ has been scheduled for the cycle. Then $wv_{k'}$ is set equal to zero, and we proceed with step (a).

If $d_{k'} > 0$, then the course section $A_{k'}$ has not been scheduled for the cycle. Replace the units per day counter with the units per day constant.

$$u_{k'} = uc_{k'}$$

Proceed with step (a).

Function 4 is repeated until we have exhausted all course sections for which $cv2_k = 1$. Then all the forced course sections have been scheduled in the i-th period, and the program control is passed to Function 2 to continue to schedule all course sections in the i-th period for which the constraints are not violated.

It should be noted that the course sections scheduled by Functions 3 and 4 are the course sections which belong to the subset $WV(j) \cap DV(j) \cap AV(j) \cap CV2(j)$, for $j = 1,2,\ldots, n$.

SUMMARY

The model of an educational scheduling system was decomposed into five phases.

Phase I A- Determination Of Course Sections To Be Offered. Predicated on student demand and faculty availability, the course sections to be offered for the term are generated by the computer.

B- Administrative Changes. The results of Phase I A and the requirements are re-examined and feasible adjustments and changes are made on the number of course sections to be offered and on the maximum requirement vector.

Phase II - Assignment Of Course Sections To Faculty, Students, And Physical Plant. These specific course section assignments are used in the development of the Relations Matrix in the next phase. When multiple sections exist, assignment of a particular section to a student may be accomplished in various ways, such as homogeneous grouping based on scholastic ability, faculty preference of a student, specific percentage of students from different departments, and balanced sections. The subset of course sections, which is assigned to a student, a faculty member, or a physical plant facility, is called a request block.

Phase III - Development Of The Constraint Structures Imposed On The System. Using the request blocks developed in Phase II, a Relations Matrix is constructed. With this matrix it is possible to determine the course sections which can be scheduled together during the same period. The number of constraints imposed on each course section is also determined.

A partially ordered Descending Chain of course sections is structured.

Phase IV - Establishment Of Priorities Based On The System Constraints And The Course Specifications. From the Descending Chain and designated course specifications, five partially ordered subsets of course sections are generated. The subsets correspond to the number of days a course section meets per cycle. This arrangement of subsets and course sections within the subsets is the priority list.

Phase V - Construction Of The Master Schedule To Satisfy The Conditions Established In Phase I Through IV. The Master Schedule is structured from the Relations Matrix and the Priority List determined in previous phases. The computer program accomodates two independent sessions per day, with provisions made for consecutive period meetings of a course section. An iterative procedure is used to develop the Master Schedule. The procedure starts with the first period of the first schedule day, repeating for each successive period until all course sections have been scheduled, without violating any constraints of the scheduling system. The number of periods so obtained to schedule all course sections is minimal for the assignment policy used for multi-section courses.

It is indeed possible that the number of periods required exceeds the number of periods permitted for a cycle. For example, if classes are scheduled 8 periods per day, five days per week, then the total schedule cycle is 40 periods. This means that the non-conflicting schedule must be forced into a cycle not to exceed the maximum number of periods permitted. If the non-conflicting schedule cycle exceeds this maximum number, then the constraints of the scheduling system must be relaxed.

A criterion must be established to permit this relaxation of restrictions. In the following report a criterion is selected to relax the constraints of the scheduling system which exceeds the maximum number of periods in a cycle.

CONTROLLED SCHEDULE CYCLE

BY

I.B. TURKSEN

A.G. HOLZMAN

INTRODUCTION

The earlier research studies of this project placed in focus the elements in the educational scheduling system. The decomposition and analysis of the scheduling resulted in the formulation of criteria for optimality.

Resultant criteria are:

1. No conflicts permitted for elements of the system,
2. Master schedule must have a minimum number of days per cycle.

It is presupposed that the specifications of the course sections are strictly observed. A specific algorithm was designed to generate a Master Schedule in compliance with the stated criteria.

Sample output resulted in Master Schedules which require more than five days. Since, traditionally, a master schedule is usually fitted into a week (5 or 6 days)*, bilateral objectives of optimality and practicality must be considered. Two alternative approaches can be taken so that the dual objectives are met. Either the weekly schedule cycle is challenged, or the criteria of optimality is relaxed.

If the concept of the weekly schedule cycle is challenged, then it is necessary to investigate the optimum length of a schedule cycle, which would be an extensive research project itself. However, if the concept of the weekly schedule is accepted, then the criteria must be relaxed.

*Our experience with high schools has revealed that some senior high schools are using essentially a one day cycle.

CRITERIA REVISIONS

Relaxation resulted in the following revised criteria:

1. Minimum number of conflicts permitted,
2. Master schedule must not require more than a specified number of days in a schedule cycle.

These revisions require an extension of the construction algorithm. This extension will be discussed later. It is desirable now to distinguish between the original and the revised criteria.

Consider a scheduling system where; 1) the length of the schedule cycle is not to exceed five days; 2) a course section A_k requires four days per cycle; and 3) the constraints of the system are such that A_k can only be scheduled with conflicts in the first and second days of the schedule cycle.

The original criteria do not allow any conflicts, but searches for the optimum cycle length. With these criteria, A_k cannot be scheduled in the first or second day of the cycle. The schedule cycle is thus forced beyond five days, since now, A_k, can at best be scheduled the third, fourth, fifth, and sixth days of the cycle.

The revised criteria does demand that the schedule cycle be fixed, but permits a minimum number of conflicts. In the example cited here, the schedule cycle is fixed at five days. Course section A_k is forced into the schedule of the second day with the units of A_k scheduled in the periods which create the minimum number of conflicts. The forcing of course sections may occur at different stages of the scheduling cycle; however, in this research, at the end of each schedule day, course sections are screened to insure that the requirements of the revised criteria are met.

In each day of the cycle, course sections are scheduled in compliance with the original criteria. If the screening indicates that one or more course sections must be forced into the schedule of this day, then the indicated course sections are scheduled in periods of this day so that conflicts are minimized. The conflicts introduced by these forced course sections are also indicated.

CONTROLLED SCHEDULE CYCLE

A familiarity with the concepts and symbols developed in the pervious papers is assumed. Development of the required new concepts follows.

A parameter ND (number of days) designates the exact length of the schedule cycle. This parameter ND usually has a value of five or six. Elements ndr_i of a vector NDR (number of days remaining) designate the number of days from the start of the i-th day to the end of the cycle. If ND = 5, then $ndr_1 = 5$, $ndr_2 = 4$, $ndr_3 = 3$, $ndr_4 = 2$, $ndr_5 = 1$, where ndr_i corresponds to the i-th day. If ND = 6, then $ndr_1 = 6$ for the first day, and $ndr_6 = 1$ for the last day of the cycle.

A set F_i designates the collection of course sections that must be screened at the end of the i-th schedule day. In a schedule cycle of five days;

F_1 contains the course sections of the first subset of the Priority List.

F_2 is the union of the first and second subsets of the Priority List

F_3 is the union of the first, second, and third subsets

F_4 is the union of the first, second, third, and fourth subsets

F_5 contains all the course sections of the Priority List.

In a schedule cycle of six days,

F_1 is a null set*

F_2 contains the course sections of the first subset of the priority list

F_3 is the union of the first and second subsets

.
.
.

F_6 contains all the course sections of the Priority List.

It should be recalled that a course section A_k has a days per cycle specification d_k. Furthermore, the days per cycle specification d_k is reduced by one whenever A_k is scheduled for a day. Therefore, d_k also indicates the number of days required by A_k for further scheduling. At the end of the i-th day, d_k is compared with ndr_i for each A_k that belongs to F_i. If $d_k = ndr_i$, then the course section A_k is forced to be scheduled. If, however, $d_k < ndr_i$, then A_k is not scheduled. The condition where $d_k > ndr_i$ can never exist since the first day of cycle $ndr_1 \geq d_k$, for $k = 1, 2, \ldots, n$.

The periods of the i-th day are scanned in order to find suitable u_k consecutive periods for each A_k that has

---

*It is assumed that no course section requires more than 5 days per cycle.

$d_k = ndr_i$. For each j-th period of the i-th day, the course sections already scheduled to the j-th period are checked to establish the number of conflicts that would result if a unit of Λ_k were scheduled in the j-th period.

To facilitate these checks, three vectors are introduced. An element of p_j of P, called the Period Vector is associated with the j-th period of a schedule day.

Period Vector	Periods of a Day
$p_1 = 1$	First period
$p_2 = 2$	Second period
$\vdots$	$\vdots$
$p_j = j$	j-th period
$\vdots$	$\vdots$
$p_m = m$	m-th period

m is the maximum number of periods in a day and equals parameter I defined previously. The elements of the Number of Conflicts Vector, NC, and the Number of No Conflicts Vector, NNC, are associated with the periods of a day, also.

Period Vector	Number of Conflicts	Number of No Conflicts
$p_1 = 1$	nc_1	nnc_1
$p_2 = 2$	nc_2	nnc_2
$\vdots$	$\vdots$	$\vdots$
$p_j = j$	nc_j	nnc_j
$\vdots$	$\vdots$	$\vdots$
$p_m = m$	nc_m	nnc_m

An element nc_j of NC indicates the number of conflicts that would result if A_k were scheduled in the j-th period. An element nnc_j of NNC indicates the number of course sections that would not be in conflict with A_k if A_k were scheduled in the j-th period.

After NC and NNC vectors are found the elements of P vector are sorted with respect to the elements of NC and NNC. For each pair p_j and p_j',

p_j is ordered before p_j' if $nc_j < nc_j'$, and if $nc_j = nc_j'$, then p_j is ordered before p_j' only if $nnc_j > nnc_j'$.

This arrangement of the symbolic names of the periods indicates the priority standing of the periods for the course section A_k. From this Period Priority Listing of periods suitable u_k consecutive periods are found and A_k is scheduled to those u_k consecutive periods. The bookkeeping is traced as described previously.

THE EXTENSION TO THE ALGORITHM

If F_i is empty for the i-th day no forced scheduling is necessary. Start the schedule of day i + 1.

If F_i is not empty, then screen the course sections of F_i.

(a) Take the next course section A_k of F_i in the order of the priority list.

(b) If, for A_k, $d_k < ndr_i$, then A_k need not be scheduled. Start with (a) again.

If $d_k = ndr_i$, then A_k must be scheduled.

(c) Scan the schedule of i-th day. For each period j, find nc_j and nnc_j

(d) Sort p_j with respect to nc_j and nnc_j and form the priority listing of the periods.

(e) If $u_k = 1$, then schedule A_k to the period with the highest priority.

If $u_k > 1$, then search the Period Priority Listing. This search is made in order to find u_k consecutive high priority periods, either in the morning or in the afternoon session, to accommodate the u_k units of the course section A_k.

(f) Do the bookkeeping of u_k, d_k, dv_k, and wv_k as described previously. Indicate the conflicts that are caused by this schedule.

(g) Start with (a) again until all the course sections in F_i are checked.

(h) Display the revised schedule of the i-th day. Start the schedule of day i + 1.

SUMMARY

The consideration of optimality and practicality has led to the development of a controlled schedule cycle. New concepts were developed and an extension to the construction algorithm was specified. A two stage procedure is used. First, an attempt is made to schedule course sections free of conflict for a particular day. Second, if for this day, it is not possible to schedule all course sections free of conflict, then the course sections with conflicts are forced into the schedule. The forcing criteria are developed to minimize the number of conflicts in the schedule for each day of the schedule cycle. The generalized Flow Chart of the system is shown in Figure 1.

FLOW CHART

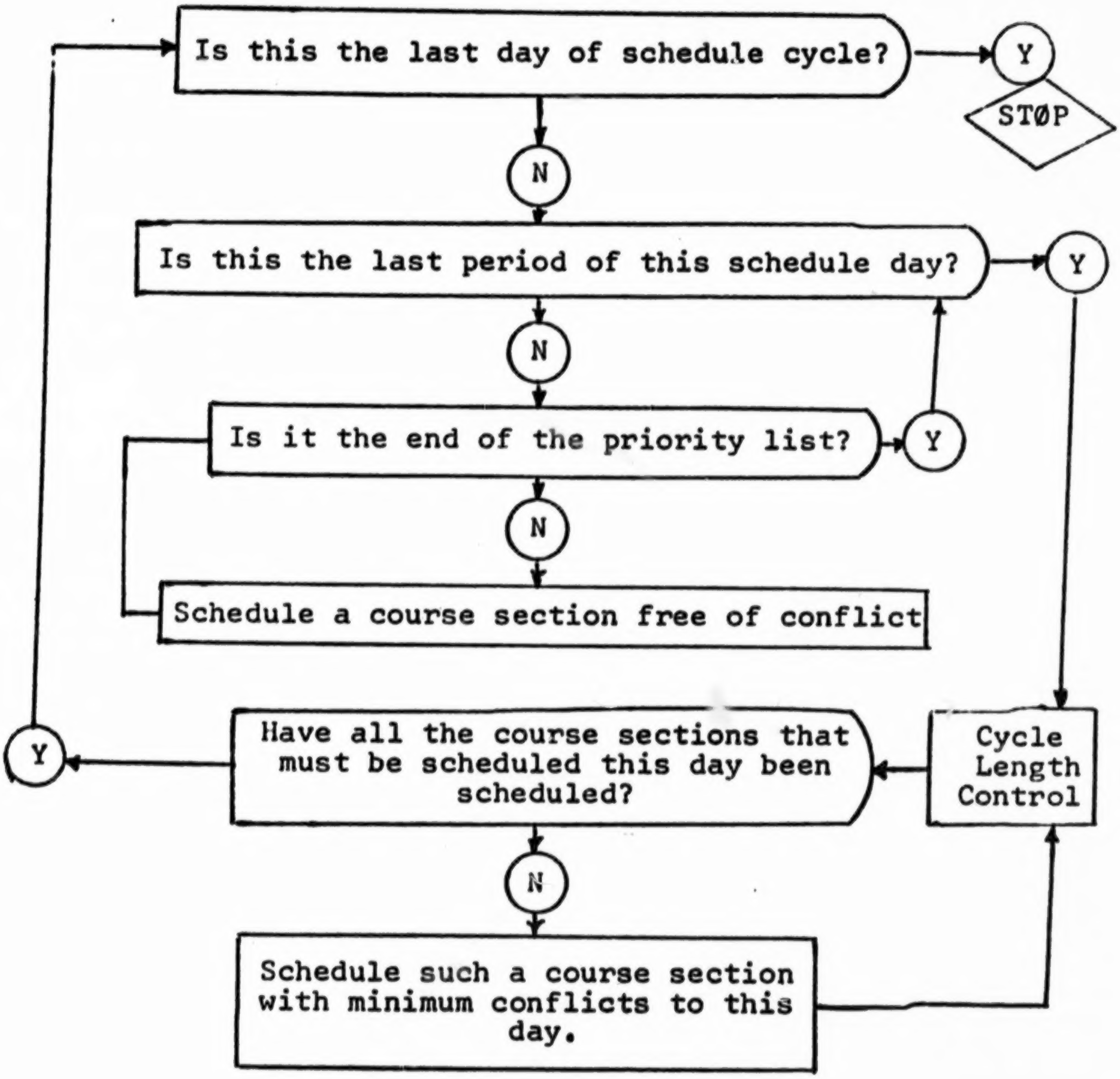

Is this the last day of schedule cycle?
Y
STØP
N
Is this the last period of this schedule day?
Y
N
Is it the end of the priority list?
Y
N
Schedule a course section free of conflict
Y
Have all the course sections that must be scheduled this day been scheduled?
Cycle Length Control
N
Schedule such a course section with minimum conflicts to this day.

Figure 1

DOCUMENTATION

OF

MASTER SCHEDULE PROGRAMS

BY

I.B. TURKSEN

A.G. HOLZMAN

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Refer to Figure 1 for the write-up of the program.

This program requires three sets of input. The sets should be arranged as indicated.

a. Parameters

b. Request blocks

c. Course Section Specification List

An explanation of each set follows.

a. Parameters. Parameters to the system are supplied by a single card.

Columns	Parameters
1- 6	NCSS
7-12	IPD
13-18	IPM
19-24	IPA
25-30	NRB

The association of the symbolic parameters with the concepts mentioned in other reports by Holzman and Turksen, follows:

NCSS: Number of course sections. If A_n is the last course section of A-Identifier list, then NCSS = n

IPD: Periods per day. This was indicated by I.

IPM: Periods per morning session. This was indicated by J.

IPA: Periods per afternoon session. This was indicated by J'.

NRB: Number of request blocks. This is the total number of request blocks submitted for student, faculty, and physical plant facility assignments.

b. Request Blocks. The number of cards in this set is equal to NRB. Each card contains the A-Identifiers of the course sections assigned to a faculty, a student or a physical plant facility. Each card contains at most 14 A-Identifiers. It is assumed that an A-Identifier is a number between 1 and 99 999. Each five column contains an A-Identifier from left to right. Each A-Identifier is right justified within its 5 column field.

Columns	Course Sections	
1- 5	First	A-Identifier
6-10	Second	"
.	.	
.	.	
.	.	
66-70	Fourteenth	"

The remaining ten columns of each request block card may be used for the identification purposes. This identification may be utilized to indicate the student, faculty member, or facility, associated with this particular request block. This identification has not been developed as yet.

c. Course Section Specification List. The number of cards in this set equals NCSS. Each card of this set contains course section specifications for a course section A_i.

Column	Program Symbols	Specifications
1- 6	ISN	A_i
7-12	IDPW	Days per cycle
13-18	IUPD	Units per day
19-24	IBD	Buffer day requirement

Internally, units per day constant is generated from units per day specification.

Refer to Figure 2 for the visual form of the three sets of inputs.

The outputs of this program displays:

a. Relations Matrix
b. Priority List
c. Master Schedule

The concepts for each output have been explained previously. Refer to Figure 3 for visual form of outputs. The last page of the output represents three sets of indicators. The first set indicates, (1) the number of days required for the non-conflicting Master Schedule and (2) the number of periods used in this last day. The second set indicates that all the course sections in the system have been scheduled, or have not been scheduled. If all have been scheduled then this vector contains all zeros. The third indicator shows that this was a normal return to the monitor system of the computer. This indicator reads 1641. If a number other than 1641 appears, then this is an indication that an error has occurred.

PROGRAM OF CONTROLLED SCHEDULE CYCLE

Refer to Figure 4 for the write-up of the program.

The sets of input to this program are exactly the same as before. Refer to Figure 5.

a. Parameters

b. Request blocks

c. Course Section Specifications

However, the parameter card has one additional parameter. The columns 31-36 of this parameter card contains the number of days required in the controlled cycle of the master schedule. This parameter has NDRP as a symbol.

The output of this program is partially similar to the output of the previous program.

a. Relations Matrix

b. Priority List

c. Master schedule

For each schedule day i, a set of results are indicated under Section c.

1. Master Schedule of day i (non-conflicting)
2. List of conflicts that must be allowed
3. Revised Master Schedule of day i.

Refer to Figure 6 for visual form of the output.

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1

```
Z
 TURKSEN           100016             DECK
                          RINDUE 00395                    045   100   000
 $EXECUTE,DUMP
 $ASSEMBLE,PUNCH OBJECT
 *0943      TURKSEN      PITT
 *      XEQ
 *      LABEL
 *       FAP
         ENTRY    IZERO
 IZERO   SXA      X2,2
         CLA      1,4
         STA      CR
         CLA*     2,4
         PDX      **,2
         TXI      *+1,2,-1
         SXD      TEXT,2
         AXT      0,2
 CR      STZ      **,2
         TXI      *+1,2,1
 TEXT    TXL      *-2,2,**
 X2      AXT      **,2
         TRA      3,4
         END
 $ASSEMBLE,PUNCH OBJECT
 *      LABEL
 *       FAP
         ENTRY    IONES
 IONES   SXA      X2,2
         CLA      1,4
         STA      CR
         CLA*     2,4
         PDX      **,2
         TXI      *+1,2,-1
         SXD      TEXT,2
         AXT      0,2
         ZAC
         COM
 CR      SLW      **,2
         TXI      *+1,2,1
 TEXT    TXL      *-2,2,**
 X2      AXT      **,2
         TRA      3,4
         END
 $ASSEMBLE,PUNCH OBJECT
 *      LABEL
 *       FAP
         ENTRY    IPSIG
 IPSIG   STZ*     2,4
         CAL*     1,4
         TZE      *+2
         STL*     2,4
         TRA      3,4
         END
 $ASSEMBLE,PUNCH OBJECT
 *      LABEL
 *       FAP
         ENTRY    SCONS
 SCONS   SXA      XR2,2
         CLA      1,4
         STA      X
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
        CLA     2,4
        STA     Y
        CLA     3,4
        STA     Z
        LDQ     =O777777777776
        STQ     XSET
        LDQ     =1
        STQ     YSET
        STZ     ZSET
        AXT     0,2
        CAL     =O777777777777
LOOP    LDQ     XSET
        RQL     35
        STQ     XSET
X       STQ     **,2
        LDQ     YSET
        RQL     35
        STQ     YSET
Y       STQ     **,2
        LDQ     ZSET
        LGR     1
        STQ     ZSET
Z       STQ     **,2
        TXI     *+1,2,1
        TXL     LOOP,2,35
XR2     AXT     **,2
        TRA     4,4
XSET    PZE
YSET    PZE
ZSET    PZE
        END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RTRM(K,KK,MM)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
      DO 1054 J=K,KK
 1054 READ TAPE MRTAPE,(R(L),L=1,MM)
      RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RRMO(KG,I,MM)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
      DO 1810 J=KG,I
 1810 READ TAPE MRT1,(R(L),L=1,MM)
      IF(I-NRTO)4001,4002,4002
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
 4002 REWIND MRT1
      KG=1
       GO TO 4005
 4001 KG=I+1
 4005 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RRMT(KG,I5,MM)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),W((100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
      DO 1815 J=KG,I5
 1815 READ TAPE MRT2,(R(L),L=1,MM)
      IF(I-NCSS)4001,4002,4002
 4002 REWIND MRT2
      KG=1
      IT=0
      GO TO 4005
 4001 KG=I5+1
 4005 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
       SUBROUTINE LOCI(ISNI,I,NCSS)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),W((100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (1Y),ISYMC(10,100),ISP(4)
      DO 2145 L=I,NCSS
       IF( ISNI-ISN(L))2145,2146,2145
 2145 CONTINUE
 2146 I=L
      RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RVIB(I,MRW,MRP,MM,IS)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
      ISYM(NP,2)=ISN(I)
 1801 IT=0
      IF(I-NRTO)1806,1806,1808
 1806 REWIND MRT2
      KG=1
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
      CALL RRMO(KG,I,MM)
      GO TO 1819
 1808 REWIND MRT1
      IT=1
      KG=1
      I5=I-NRTO
      CALL RRMT (KG,I5,MM)
 1819 DO 1820 L=1,MM
B     RE(NP,L)=R(L)*WV(L)
B     RE(NP,L)=RE(NP,L)*DV(L)
B1820 RE(NP,L)=RE(NP,L)*AV(L)
      IUPD(I)=IUPD(I)-1
      IF(IUPD(I))1826,1830,1840
 1826 ISTOP=1626
      WRITE OUTPUT TAPE IOTAPE,1827,ISTOP
 1827 FORMAT(1H ,I6)
      CALL ERROR
 1830 CALL UPDZ(I,MRW,MRP)
      GO TO 1850
B1840  CV1(MRW)= CV1(MRW)+ Y(MRP)
      IFNP=1
 1850 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE UPDZ(I,MRW,MRP)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
   11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
   12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
   13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(35),Z(35),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (1Y),ISYMC(10,100),ISP(4)
B      DV(MRW)= DV(MRW)* X(MRP)
      IDPW(I)=IDPW(I)-1
      IF(IDPW(I))1855,1860,1870
 1855 ISTOP=1855
      WRITE OUTPUT TAPE IOTAPE,1856,ISTOP
 1856 FORMAT(1H ,I6)
      CALL ERROR
B1860  AV(MRW)= WV(MRW)* X(MRP)
      GOTO 1880
 1870 IUPD(I)=IUPDC(I)
 1880 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE JTHC(I,MRW,MRP,NRPS,MM,IS)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
   11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
   12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
   13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
B      PC=RE(NP,MRW)*Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1950,1940,1950
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
 1940 GO TO 2060
 1950 IF(IUPD(I)-NRPS)1960,1960,1980
 1960 IF(IDB(I))1935,2000,1990
 1935 ISTOP=1935
      WRITE OUTPUT TAPE IOTAPE,1936,ISTOP
 1936 FORMAT(1H ,I6)
      CALL ERROR
B1980  AV(MRW)= AV(MRW)* X(MRP)
      GO TO 2060
B1990  PC= DB1(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1995,1830,1995
B1830 RE(NP,MRW)=RE(NP,MRW)*X(MRP)
      GO TO 2060
B1995  DB2(MRW)= DB2(MRW)* X(MRP)
 2000 IS=IS+1
      ISYM(NP,IS)=ISN(I)
 2001 IF(IT)1935,2002,2010
 2002 IF(I-NRTO)2005,2005,2004
 2005 CALL RRMO(KG,I,MM)
      GO TO 2019
 2004 REWIND MRT1
      IT=1
      KG=1
      I5=I-NRTO
      CALL RRMT(KG,I5,MM)
      GO TO 2019
 2010 I5=I-NRTO
      CALL RRMT(KG,I5,MM)
 2019 DO 2020 L=1,MM
B2020 RE(NP,L)=RE(NP,L)*R(L)
      IUPD(I)=IUPD(I)-1
      IF(IUPD(I))1935,2040,2050
 2040 CALL UPDZ(I,MRW,MRP)
      GO TO 2060
B2050  CV1(MRW)= CV1(MRW)+ Y(MRP)
      IFNP=1
 2060 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE ORDER
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
    13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
     3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
 3130 KE=K-1
      KB=K-ISAME
      KBB=KB+1
      DO 3150 LL=KB,KE
      DO 3150 L=KBB,KE
      IF(IFDNC(L)-IFDNC(L-1))3150,3150,3140
 3140 ITEM=IFDNC(L-1)
      ITEN=IFDC(L-1)
      ITET=IFDP(L-1)
      IFDNC(L-1)=IFDNC(L)
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
          IFDC(L-1)=IFDC(L)
          IFDP(L-1)=IFDP(L)
          IFDNC(L)=ITEM
          IFDC(L)=ITEN
          IFDP(L)=ITET
 3150     CONTINUE
          ISW=0
          ISAME=1
          RETURN
          END
$COMPILE FORTRAN,PUNCH OBJECT
*         LABEL
C               MAIN PROGRAM
        1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
       11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
       12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT,
       13KE,KB,ISAME,KBB,LL,L,IFDNC,IFDC,IFDP,ITEM,ITEN,ITET,ISW
          DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000[illegible]UPD(10
         100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
         2),DB1(100),DB2(100),R(100),RE(10,100),ISYM(10,200),IRBS(14),
         3IFDP(10),IFDNC (10),IFDC (10),ISYMC(10,100),ISP(4)
          IRTAPE=3
          ISTAPE=2
          MRT2=2
          MRT1=3
          MRTAPE=9
          ITAPE=7
          IOTAPE=6
          CALL SCONS(X,Y,Z)
          READ INPUT TAPE  ITAPE,15,NCSS,IPD,IPM,IPA,NRB,NDRP
   15     FORMAT(6I6)
          REWIND IRTAPE
          REWIND ISTAPE
          MM=NCSS/36
          MMP=NCSS-MM*36
          IF(MMP)90,16,18
   16     MMP=36
          MMM=MM
          GO TO 19
   18     MMM=MM
          MM=MM+1
          NBO=NRB/2
          NBT=NRB-NBO
   19     DO 50 J=1,NBO
          READ INPUT TAPE  ITAPE,51,(IRBS(L),L=1,14)
   51     FORMAT(14I5)
   50     WRITE TAPE IRTAPE,(IRBS(L),L=1,14)
          END FILE IRTAPE
          REWIND IRTAPE
          DO 12 J=1,NBT
          READ INPUT TAPE  ITAPE,51,(IRBS(L),L=1,14)
   12     WRITE TAPE ISTAPE,(IRBS(L),L=1,14)
          END FILE ISTAPE
          REWIND ISTAPE
          DO 20 J=1,NCSS
   20     ICC(J)=0
          REWIND MRTAPE
          WRITE OUTPUT TAPE IOTAPE,35
   35     FORMAT(1H1/52X,16HRELATIONS MATRIX//50X,18HELEMENTS ARE OCTAL//55X
   351,8HDECOD AS//57X,5H0=000/57X,5H1=001/57X,5H2=010/57X,5H3=011/57X,5
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
352H4=100/57X,5H5=101/57X,5H6=110/57X,5H7=111//24X,3HROW//)
 21 DO 300 K=1,NCSS
 30 REWIND IRTAPE
    REWIND ISTAPE
    CALL IONES (R,MM)
    DO 200 J=1,NRB
    IF(J-NBO)9,9,8
  9 REWIND ISTAPE
    READ TAPE IRTAPE,(IRBS(L),L=1,14)
    GO TO 5
  8 REWIND IRTAPE
    READ TAPE ISTAPE,(IRBS(L),L=1,14)
  5 DO 70 L=1,14
    IF(K-IRBS(L))70,60,70
 70 CONTINUE
    GO TO 200
 60 IRBS(L)=0
    DO 80 L=1,14
    IF(IRBS(L))90,80,100
 90 ISTOP=90
 91 WRITE OUTPUT TAPE IOTAPE,95
 95 FORMAT(1H0,35HDATA IS BAD.LOOK AT R-ARRAY IN DUMP)
    GO TO 5000
100 ICC(K)=ICC(K)+1
    MRW=IRBS(L)/36
    MRP=IRBS(L)-MRW*36
    IF(MRP)90,120,130
120 MRP=36
    GO TO 135
130 MRW=MRW+1
135 IF(MRW-100)140,140,150
150 ISTOP=150
    GO TO 91
B 140  R(MRW)= R(MRW)* X(MRP)
 80 CONTINUE
200 CONTINUE
    WRITE TAPE MRTAPE,(R(L),L=1,MM)
    WRITE OUTPUT TAPE IOTAPE,210,K,(R(L),L=1,MM)
210 FORMAT(1H ,20X,I6,3X,7012)
300 CONTINUE
    REWIND IRTAPE
    REWIND ISTAPE
    END FILE MRTAPE
    REWIND MRTAPE
    REWIND ISTAPE
    DO 53 J=1,NCSS
    READ INPUT TAPE ITAPE,52,ISN(1),IDPW(1),IUPD(1),IDB(1)
 52 FORMAT(4I6)
 53 WRITE TAPE ISTAPE,ISN(1),IDPW(1),IUPD(1),IDB(1)
    END FILE ISTAPE
    REWIND ISTAPE
    DO 310 K=1,NCSS
310 ISN(K)=K
    DO 400 K=1,NCSS
    DO 500 L=2,NCSS
    IF(ICC(L)-ICC(L-1))500,500,450
450 ITEM=ICC(L-1)
    ITEN=ISN(L-1)
    ICC(L-1)=ICC(L)
    ISN(L-1)=ISN(L)
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
      ICC(L)=ITEM
      ISN(L)=ITEN
  500 CONTINUE
  400 CONTINUE
      REWIND ISTAPE
      DO 900 K=1,NCSS
      READ TAPE ISTAPE,(ISP(J),J=1,4)
      DO 800 I=1,NCSS
      IF(ISN(I)-ISP(1))800,850,800
  850 IDPW(I)=ISP(2)
      IUPD(I)=ISP(3)
      IUPDC(I)=ISP(3)
      IDB(I)=ISP(4)
      GO TO 900
  800 CONTINUE
      WRITE OUTPUT TAPE IOTAPE,860,(ISP(J),J=1,4)
  860 FORMAT(1H0,15HBAD INFORMATION,4I6)
  900 CONTINUE
      DO 910 L=1,NDRP
  910 IRBS(L)=0
      REWIND ISTAPE
      K=NDRP
      WRITE OUTPUT TAPE IOTAPE,941
  941 FORMAT(1H1,53X,14HPRIORITY  LIST)
      WRITE OUTPUT TAPE IOTAPE,940
  940 FORMAT(1H0,9X,6HCOURSE,7X,8HCONFLICT,7X,8HDAYS PER,6X,9HUNITS PER,
 94016X,9HUNITS PER,9X,6HBUFFER/8X,8HSECTIONS,8X,7HCOUNTER,10X,5HCYCLE,
 940212X,3HDAY,15H   DAY CONSTANT,15H    REQUIREMENT/)
  950 DO 1000 I=1,NCSS
      IF(K-IDPW(I))1000,1010,1000
 1010 WRITE TAPE ISTAPE,ISN(I),ICC(I),IDPW(I),IUPD(I),IUPDC(I),IDB(I)
      WRITE OUTPUT TAPE IOTAPE,1005,ISN(I),ICC(I),IDPW(I),IUPD(I),
     1IUPDC(I),IDB(I)
 1005 FORMAT(1H ,6I15)
      IRBS(K)=IRBS(K)+1
 1000 CONTINUE
      K=K-1
      IF(K)90,1030,950
 1030 END FILE ISTAPE
      REWIND ISTAPE
      WRITE OUTPUT TAPE IOTAPE,1075,(IRBS(K),K=1,NDRP)
 1075 FORMAT(1H ,10I6)
      DO 1050 I=1,NCSS
 1050 READ  TAPE ISTAPE,ISN(I),ICC(I),IDPW(I),IUPD(I),IUPDC(I),IDB(I)
      REWIND ISTAPE
 1055 NRTO=NCSS/2
      REWIND MRTAPE
      REWIND MRT1
      REWIND MRT2
      KK=ISN(1)
      K=1
      CALL RTRM(K,KK,MM)
      WRITE TAPE MRT1,(R(L),L=1,MM)
      DO 1080 I=2,NCSS
      IF(ISN(I)-ISN(I-1))1070,1069,1060
 1069 ISTOP=1069
      GO TO 5000
 1070 REWIND MRTAPE
      K=1
      GO TO 1063
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
 1060 K=KK+1
 1063 KK=ISN(I)
 1065 CALL RTRM(K,KK,MM)
      IF(I-NRTO)1066,1066,1068
 1066 WRITE TAPE MRT1,(R(L),L=1,MM)
      GO TO 1080
 1068 WRITE TAPE MRT2,(R(L),L=1,MM)
 1080 CONTINUE
      END FILE MRT1
      END FILE MRT2
      REWIND MRTAPE
      REWIND MRT1
 1085 REWIND MRT2
      WRITE OUTPUT TAPE IOTAPE,5010
 5010 FOPMAT(1H1,52X,15HMASTER SCHEDULE)
      CALL IONES(WV,MM)
      CALL IONES(DB2,MM)
      IF(MMP-36)1152,1153,1390
31152  WV(MM)= WV(MM)* Z(MMP)
 1153 CALL IZERO(CV1,MM)
      CALL IZERO(CV2,MM)
      NCSDS=0
      NDR=NDRP+1
      ND=0
 1545 IF(ND-NDRP)3300,3400,3400
 3300 ND=ND+1
      NDR=NDR-1
 1690 WRITE OUTPUT TAPE IOTAPE,1700 ,ND
 1700 FORMAT(1H1,46X,22HMASTER SCHEDULE DAY    ,I6//6HPERIOD/)
      CALL IONES(DV,MM)
      CALL IONES(AV,MM)
      DO 1550 J=1,MM
 1550 DB1(J)=DB2(J)
      CALL IONES(DB2,MM)
 1160 IPS=IPM
      NP=1
      NPS=0
      NRPS=IPS
      IMA=1
      IAOS=0
      IFNP=0
 1309 REWIND MRT1
      REWIND MRT2
      DO 1800 L=1,200
 1800 ISYM(NP,L)=0
      IS=2
      IF(IFNP)1310,1320,2100
 1310 ISTOP=1310
      GO TO 5000
 1320 REWIND MRT1
      REWIND MRT2
      DO 1810 L=1,200
 1810 ISYM(NP,L)=0
      IS=2
      CALL IONES(R,MM)
      DO 1820 L=1,MM
 1820 RE(NP,L)=R(L)
      DO 1560 I=1,NCSS
 1321 IF(ISN(I))1310,1310,1340
 1340 MRW=ISN(I)/36
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
      MRP=ISN(I)-MRW*36
      IF(MRP)1310,1360,1370
 136  MRP=36
      GO TO 1375
 1370 MRW=MRW+1
 1375 IF(MRW-MM)1380,1380,1390
 1390 ISTOP=1390
      GO TO 5000
B1380  PC= WV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1420,1560,1420
B1420  PC= DV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1430,1560,1430
B1430  PC= AV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1440,1750,1440
 1750 IAOS=1
      GO TO 1560
 1440 IF(IUPD(I)-NRPS)1450,1450,1520
B1520  AV(MRW)= AV(MRW)* X(MRP)
      GO TO 1750
 1450 IF(IDB(I))1310,1470,1510
B1510  PC= DB1(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1512,1830,1512
B1830 RE(NP,MRW)=RE(NP,MRW)*X(MRP)
      GO TO 1560
B1512  DB2(MRW)= DB2(MRW)* X(MRP)
      GO TO 1470
 1560 CONTINUE
 3400 DO 1564 J=1,MM
      PC=WV(J)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1565,1564,1565
 1564 CONTINUE
 1510 WRITE OUTPUT TAPE IOTAPE,1620,ND,NP
 1620 FORMAT(1H1,15HNUMBER OF DAYS=,I6,5X,7HPERIOD=,I6)
      WRITE OUTPUT TAPE IOTAPE,1640,(WV(L),L=1,MM)
 1640 FORMAT(1H0,100I2)
 1641 ISTOP=1641
      WRITE OUTPUT TAPE IOTAPE,1642,ISTOP
 1642 FORMAT(1H0,I6)
      CALL SYSTEM
 1546 DO 1547 J=NP,IPD
      DO 1548 L=1,MM
 1548 RE(J,L)=0
      DO 1549 L=1,200
 1549 ISYM(J,L)=0
 1547 ISYM(J,1)=2
      GO TO 3000
 1565 IF(IMA)1310,1546,1570
 1570 IMA=0
      IF(IAOS)1310,1546,1651
 1651 DO 1652 J=NP,IPS
       DO 1653 L=1,MM
 1653 RE(J,L)=0
      DO 1652 L=1,200
 1652 ISYM(J,L)=0
 1650 WRITE OUTPUT TAPE IOTAPE,1660
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
 1660 FORMAT(1H0,51X,18H AFTERNOON SESSION)
 1720 IPS=IPA
      IMA=0
      IF(IPS)1310,3000,1730
 1730 NPS=0
      NRPS=IPS
      CALL IONES(AV,MM)
      GO TO 1320
 1470 CALL RVIB(I,MRW,MRP,MM,IS)
      II=I+1
 1475 DO 1480 I=II,NCSS
      MRW=ISN(I)/36
      MRP=ISN(I)-MRW*36
      IF(MRP)1310,1890,1900
 1890 MRP=36
      GO TO 1905
 1900 MRW=MRW+1
 1905 IF(MRW-MM)1910,1910,1390
B1910  PC= CV2(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1480,1930,1480
 1930 CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 1480 CONTINUE
      ISYM(NP,1)=IS
 1490 WRITE OUTPUT TAPE IOTAPE,1530,NP,(ISYM(NP,J),J=2,IS)
 1530 FORMAT(1H ,I5,3X,18I6/(9X,18I6))
      NP=NP+1
      IF(NP-IPD)1710,1710,3000
 1710 NPS=NPS+1
      NRPS=IPS-NPS
      IF(NRPS)1310,1650,1309
 2100 IFNP=0
      DO 2110 J=1,MM
 2110 CV2(J)=CV1(J)
      CALL IZERO(CV1,MM)
      DO 2130 J=1,MM
      DO 2130 K=1,36
B      PC= CV2(J)* Y(K)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2140,2130,2140
 2130 CONTINUE
 2135 ISTOP=62135
 5000 WRITE OUTPUT TAPE IOTAPE,2136,ISTOP
 2136 FORMAT(1H ,I6)
      CALL ERROR
 2140 MRW=J
      MRP=K
      ISNI=(MRW-1)*36+MRP
      I=1
      CALL LOCI(ISNI,I,NCSS)
      CALL RVIB(I,MRW,MRP,MM,IS)
      IF(K-36)2150,2180,2180
 2150 K=K+1
      DO 2170 LL=K,36
B      PC= CV2(J)* Y(LL)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2160,2170,2160
 2160 MRW=J
      MRP=LL
      ISNI=(MRW-1)*36+MRP
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
      I=1
      CALL LOCI(ISNI,I,NCSS)
      KG=1
       IT=0
      REWIND MRT1
      REWIND MRT2
      CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 2170 CONTINUE
       IF(MM-J)1310,2231,2180
 2180 JJ=J+1
       DO 2200 J=JJ,MM
       DO 2200 K=1,36
B     PC=CV2(J)*Y(K)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2210,2200,2210
 2210 MRW=J
      MRP=K
      ISNI=(MRW-1)*36+MRP
       I=1
      CALL LOCI(ISNI,I,NCSS)
      KG=1
       IT=0
      REWIND MRT1
      REWIND MRT2
      CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 2200 CONTINUE
 2231 II=1
      REWIND MRT1
      REWIND MRT2
      IT=0
      KG=1
      GO TO 1475
 3000 NCSDS=IRBS(NDR)+NCSDS
 3319 WRITE OUTPUT TAPE IOTAPE,3058
 3058 FORMAT(1H0,33X,49HIT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICT
 30581S//7HSECTION,3X,30HCONFLICTS WITH COURSE SECTIONS/)
      DO 4000 J=1,IPD
      DO 4010 L=1,100
 4010 ISYMC(J,L)=0
 4000 ISYMC(J,1)=3
      REWIND MRT1
      REWIND MRT2
      IT=0
      KG=1
      DO 3240 I=1,NCSDS
      IF(ISN(I))1310,1310,3005
 3005 MRW=ISN(I)/36
      MRP=ISN(I)-MRW*36
      IF(MRP)1310,3010,3020
 3010 MRP=36
      GO TO 3025
 3020 MRW=MRW+1
 3025 IF(MRW-MM)3030,3030,1390
83030 PC=DV(MRW)*Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)3040,3240,3040
 3040 IF(NDR-IDPW(I))3050,3050,3240
 3050 DO 3055 IJ=1,IPD
      ISC=ISYMC(IJ,1)
      DO 3301 L=1,ISC
 3001 ISYMC(IJ,L)=0
      ISYMC(IJ,1)=3
      IFDP(IJ)=IJ
      IFDC(IJ)=0
 3055 IFDNC (IJ)=0
      IF(IT)1310,3302,3310
 3302 IF(I-NRTO)3305,3305,3304
 3305 CALL RRMO(KG,I,MM)
      GO TO 3330
 3304 REWIND MRT1
      IT=1
      KG=1
      I5=I-NRTO
      CALL RRMT(KG,I5,MM)
      GO TO 3330
 3310 I5=I-NRTO
      CALL RRMT(KG,I5,MM)
 3330 DO 3090  J=1,IPD
      ISC=3
      ISYMC(J,2) =ISN(I)
       DO 3065 L=1,MM
       IF(RE(J,L))3057,3065,3057
 3065 CONTINUE
       IFDC(J)=0
       GO TO 3086
 3057 DO 3080 L=1, MM
B     RARE=RE(J,L) *(-R(L))
      PC=RARE
      CALL IPSIG( PC,FLAG)
      IF(FLAG)3060,3080,3060
 3060 CONTINUE
      DO 3085 K=1,36
B      PC=RARE*Y(K)
      CALL IPSIG( PC,FLAG)
      IF(FLAG)3070,3085,3070
 3070 IFDC(J) =IFDC(J) +1
      ISYMC(J,ISC)=36*(L-1)+K
      ISC=ISC+1
 3085 CONTINUE
 3080 CONTINUE
 3086 ISYMC(J,1)=ISC
      IFDNC(J) =ISYM(J,1) -IFDC(J)
 3090 CONTINUE
      DO 3100 K=1,IPD
      DO 3100 L=2,IPD
      IF(IFDC(L)-IFDC(L-1))3095,3100,3100
 3095 ITEM=IFDNC(L-1)
      ITEN=IFDC(L-1)
      ITET=IFDP(L-1)
      IFDNC(L-1)=IFDNC(L)
      IFDC(L-1)=IFDC(L)
      IFDP(L-1)=IFDP(L)
      IFDNC(L)=ITEM
      IFDC(L)=ITEN
      IFDP(L)=ITET
 3100 CONTINUE
      ISW=0
      ISAME=1
      DO 3160 K=2,IPD
      IF(IFDC(K)-IFDC(K-1))3120,3110,3120
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

```
 3110 ISAME=ISAME+1
      ISW=1
      GO TO 3160
 3120 IF(ISW)3130,3160,3130
 3130 CALL ORDER
 3160 CONTINUE
      IF(ISW)3165,3168,3165
 3165 K=K+1
      CALL ORDER
 3168 IF(IUPD(I)-1)1310,3170,3180
 3170 K=IFDP(1)
B     RE(K,MRW)=RE(K,MRW)+Y(MRP)
      IS=ISYM(K,1)+1
      ISYM(K,IS)=ISN(I)
      ISYM(K,1)=ISYM(K,1)+1
      ISC=ISYMC(K,1)-1
      WRITE OUTPUT TAPE IOTAPE,1530,(ISYMC(K,J),J=2,ISC)
 3410 IUPD(I)=0
      CALL UPDZ(I,MRW,MRP)
      GO TO 3240
 3180 DO 3230 IJ=1,IPD
      K=IFDP(IJ)
      IF(K-IPM)3190,3230,3220
 3190 IAP=IPM-K+1
 3195 IF(IAP-IUPD(I))3230,3200,3200
 3200 KE=K+IUPD(I)-1
      DO 3210 IJK=K,KE
B     RE(IJK,MRW)=RE(IJK,MRW)+Y(MRP)
      IS=ISYM(IJK,1)+1
      ISYM(IJK,IS)=ISN(I)
      ISYM(IJK,1)=ISYM(IJK,1)+1
      ISC=ISYMC(IJK,1)-1
 3210 WRITE OUTPUT TAPE IOTAPE,1530,(ISYMC(IJK,J),J=2,ISC)
      GO TO 3410
 3220 IAP=IPD-K+1
      GO TO 3195
 3230 CONTINUE
 3240 CONTINUE
      WRITE OUTPUT TAPE IOTAPE,3250,ND
 3250 FORMAT(1H0,42X,30HREVISED MASTER SCHEDULE DAY    ,I6//6HPERIOD/)
      DO 3255 IJ=1,IPD
      IS=ISYM(IJ,1)
 3255 WRITE OUTPUT TAPE IOTAPE,1530,IJ,(ISYM(IJ,J),J=2,IS)
      GO TO 1545
      END
```

PROGRAM OF OPTIMAL MASTER SCHEDULE CYCLE

Figure 1 (cont.)

SAMPLE INPUT DATA

Figure 2

```
          PARAMETER CARD
192     8     8     0   538



               REQUEST BLOCK CARDS
  1    7   13   19   25   31   42   47   53   59
  1    7   13   19   25   31   37   50   53   59
  2    8   14   20   26   32   37   50   53   59
  2    8   14   20   26   32   42   47   53   59
  3    9   15   21   27   33   42   47   53   59
  3    9   15   21   27   33   37   50   53   59
  4   10   16   22   28   34   37   50   53   59
  4   10   16   22   28   34   42   47   53   59
  5   11   17   23   29   35   37   50   53   59
  5   11   17   23   29   35   42   47   53   59
 64   69   74   79   84   89   98  102  106  111
 64   69   74   79   84   89   94  104  106  111
 65   70   75   80   85   90   98  102  106  111
 65   70   75   80   85   90   94  104  106  111
 66   71   76   81   86   91   94  104  106  111
 66   71   76   81   86   91   98  102  106  111
115  124  133  154  147  162  177  181
115  124  133  158  140  162  177  187
116  125  134  154  147  166  177  187
116  125  134  158  140  166  177  181
117  126  135  158  140  162  177  181
117  126  135  154  147  166  173  187
118  127  136  158  140  166  173  181
118  127  136  158  140  162  173  187
119  128  137  158  140  162  192  181
119  128  137  154  147  191  192  187
119  128  137  154  147  166  192  181
```

SAMPLE INPUT DATA

Figure 2 (cont.)

```
COURSE SECTION SPECIFICATION LIST
             1    5    1    0
             2    5    1    0
             3    5    1    0
             4    5    1    0
             5    5    1    0
             6    5    1    0
             7    5    1    0
             8    5    1    0
             9    5    1    0
            10    5    1    0
            11    5    1    0
            12    5    1    0
            13    4    1    0
            14    4    1    0
            15    4    1    0
            16    4    1    0
            17    4    1    0
            18    4    1    0
            19    4    1    0
            20    4    1    0
            21    4    1    0
            22    4    1    0
            23    4    1    0
            24    4    1    0
            25    4    1    0
            26    4    1    0
            27    4    1    0
            28    4    1    0
            29    4    1    0
            30    4    1    0
            31    2    1    0
            32    2    1    0
            33    2    1    0
            34    2    1    0
            35    2    1    0
            36    2    1    0
            37    1    2    0
            38    1    2    0
```

OUTPUT

Figure 3

```
                         RELATIONS MATRIX

                        ELEMENTS ARE OCTAL

                             DECOD AS

                               0=000
                               1=001
                               2=010
                               3=011
                               4=100
                               5=101
                               6=110
                               7=111

 ROW

  1   573737373737363555475567777777777777777777777777777777777777777777777777
  2   375757575757161553475707777777777777777777777777777777777777777777777777
  3   706767676767065444547714777777777777777537777777777777777777777777777777
  4   647373737373224445147714777777777777777637777777777777777777777777777777
  5   627575757475306441406714777777777777777637777777777777777777777777777777
  6   517675757676756223535714777777777777777777777777777777777777777777777777
  7   375737343737363554756765777777777777777777777777777777777777777777777777
  8   573757545757161554767757777777777777777777777777777777777777777777777777
  9   577357576767654443477777777777777777777777777777777777607777777777777777
 10   736773737373224445147777777777777777777777777777777777607777777777777777
 11   757775757475306441406777777777777777777777777777777777777777777777777777
 12   757776757575756223656777777777777777777777777777777777777777777777777777
 13   373741373737363554757777777777777777777777777777777777777777777777777777
 14   575721575757161554757777777777777777777777777777777777777777777777777777
 15   676711676767654443477777777777777777777777777777777777777777777777777777
 16   737305737373224445147777777777777777777777777777777777777777777777777777
 17   757503757475306441406775777777777777777777777777777777777777777777777777
 18   767577747676756223636777770177777777777777777777777777777777777777777777
 19   373737553737363555475677607777777777777777777777777777777777777777777777
 20   575757365757161554757777607777777777777777777777777777777777777777777777
 21   676767775700065444347777777777407777777777777777777777777777777777777777
 22   737373770073224445147777777777777777777777777777777777777777777777777777
 23   751574767475306441406763770177777777777777777777777777777777777777777777
 24   761576157676756223636765607777777777777777777777777777777777777777777777
 25   373737334037363554756777777777777777777777777777777777777777777777777777
 26   575757532057161554757777777777777777777777777777777777777777777777777777
 27   676767631067065444347777777777777777777777777777777777777777777777777777
 28   737373730473224445147777777777777777777777777777777777777777777777777777
 29   757575710275306445447777777777777777777777777777777777777777777777777777
 30   747474700174756223636777777777777777777777777777777777777777777777777777
 31   373737273740363554756777777777407777777777777777777777777777777777777777
 32   575757475720161554757777777777407777777777777777777777777777777777777777
 33   676767676710065444347777777777407777777777777777777777777777777777777777
 34   737373637304224445147777777777407777777777777777777777777777777777777777
 35   757575657402306441406777777777407777777777777777777777777777777777777777
 36   767676667601756223636777777777407777777777777777777777777777777777777777
 37   010101010101417754756777777777703777777777777777777777777777777777777777
 38   474747474747217756347777777777703777777777777777777777777777777777777777
 39   636363636363117747167777777777703777777777777777777777777777777777777777
 40   717171717171057767467777777777703777777777777777777777777777777777777777
 41   747474747474037763626777777777703777777777777777777777777777777777777777
```

OUTPUT

Figure 3 (cont.)

```
 42   010101010101770175756777777777774177777777777777777777777777777777777777
 43   171717171717764174757777777777774177777777777777777777777777777777777777
 44   434343434343762076147777777777774177777777777777777777777777777777777777
 45   707070707070761277465777777777774177777777777777777777777777777777777777
 46   767676767675760637636777777777774177777777777777777777777777777777777777
 47   010101010101761634756777777777777637777777777777777037777777777777777777
 48   606060606060774136046777777777777637777777777777777037777777777777777777
 49   767676767676777077636777777777777637777777777777777037777777777777777777
 50   010101010101077760346777777777777747777777777777777741777777777777777777
 51   606060606060617753066777777777777747777777777777777741777777777777777777
 52   747474747674757747636777777777777747777777777777777741777777777777777777
 53   010101010101363556016777777777777770177777777777777777777777770777777777
 54   070707070707171455017777777777777770177777777777777777777777770777777777
 55   636363636363475644407777777777777770177777777777777777777777770777777777
 56   717171717171634664227777777777777770177777777777777777777777770777777777
 57   747474747474716220126777777777777770177777777777777777777777770777777777
 58   747474747674757330076777777777777770177777777777777777777777770777777777
 59   010101010101161454361777777777777776177777777777777777777777007777777777
 60   616161616161414647051777777777777776177777777777777777777777007777777777
 61   777777777777777777777457777777777776177777777777777777777777007777777777
 62   777777777777777777777437777777777776177777777777777777777777007777777777
 63   343434343634346001637777777777777777677777777777777777777777707377777777
 64   177777777777777777777746757367571465166777777777777777777777777777777777
 65   177777777777777777777772735756367146506777777777777777777777777777777777
 66   177777777777777777777717573675730420263777777777777777777777777777777777
 67   607777777777777777777775475736756312636777777777777777777777777777777777
 68   607777777777777777777773436757360140302777777777777777777777777777777777
 69   771777477777777777777737357367571465166777777777777777777777777777777777
 70   771777477777777777777775676756357146506777777777777777777777777777777777
 71   607777777777777777777761673675730420263777777777777777777777777777777777
 72   607777777777777777777771575736756312536777777777777777777777777777777777
 73   777777777777777777777775776757360140302777777777777777770777777757777777
 74   777701777777777777777773677367571465166777777777777777777777777777777777
 75   777777157777777777777775757056367146506777777777777777777777777777777777
 76   777777167777777777777767554675730420263777777777777777777777777777777777
 77   777777167777777777777773652736756312636777777777777777777777777777777777
 78   777777167777777777777775721757360140302777777777777777777777777777777777
 79   777767577777777777777736757407571465166777777777777777777777777777777777
 80   777767577777777777777775735720367146506777777777777777777777777777777777
 81   777767577777777777777767573115730420263777777777777777777777777777777777
 82   777767577777777777777773675056756312636777777777777777777777777777777777
 83   777767577777777777777775736037360140302777777777777777777777777777777777
 84   777777777777777777777716347170471465166777777777777777771777777777777777
 85   777777777777777777777775736756467146506777777777777777771777777777777777
 86   777777777777777777777767573662730420263777777777777777771777777777777777
 87   777777777777777777777773675721756312636777777777777777771777777777777777
 88   777777777777777777777775736757760140302777777776377777776377777777777777
 89   777776777007777777777736757367601465166777777777777777777777777777777777
 90   777776777007777777777775736756350146506777777777777777777777777777777777
 91   777776777007777777777767573675440420263777777777777777777777777777777777
 92   777776777007777777777773675736426312636777777777777777777777777777777777
 93   777776777007777777777775736757010140302777777777777777777777777777777777
 94   777777777777017777777704102041024374142777777777777777777777777777777777
 95   777777777777017777777704102041022375443777777777777777777777777777777777
 96   777777777777017777777761430614301374612777777777777777777777777777777777
 97   777777777777017777777771634716340774712777777777777777401777777777777777
 98   777777777777760377777704102041027607166777777777777777777777777777777777
 99   777777777777760377777770614306143750746777777777777777777777777777777777
100   777777777777760377777763471634717443622777777777777777777777777777777777
101   777777777777760377777771634716347433712777777777777777777777777777777777
```

OUTPUT

Figure 3 (cont.)

```
102     777777777777777437777041020410274330667777777777777037777777777777777777
103     777777777777777437777614306143077076027777777777777037777777777777777777
104     777777777777777437777041020410203760427777777777777741777777777777777777
105     777777777777777437777614306143023756127777777777777741777777777777777777
106     777777777777777740370410204102356540677777777777777777777777770777777777
107     777777777777777740371634716347146520777777777777777777777777770777777777
108     777777777777777740374306143061462010277777777777777777777777770777777777
109     777777777777777740377163471634631205677777777777777777777777770777777777
110     777777777777777740377573675736035003277777777777777777777777770777777777
111     777777777777777777741041020410214210717777777777777777777777700777777777
112     777777777777777777741655326553203106457777777777777777777777700777777777
113     777777777777777777741777777777777777637777777777777777777777700777777777
114     777777777777777777776306143061421002177777777777777777777777707377777777
115     777777777777777777777777777777777777774173773747637356736563373177777777
116     777777777777777777777777777777777777772175771743617146317563373777777777
117     777777777777777777777777777777777777771176774751647146316461371377777777
118     777777777777777777777777777777777777770577376354763446117475371377777777
119     777777777777777777777777777777777777770377577556273242416677170477777777
120     617777777777777777777777777777777777777737677657271253707664172777777777
121     617777777777777777777777777777777777777677737717074213723437172777777777
122     777777777777777777777777777777777777777777757707476311732500172177777777
123     777777777777777777777777777777777777777777767700016315730017076777777777
124     777777777777777777777777777777777777773774173747637356736563373177777777
125     777777777777777777777777777777777777775772171743617146317563373777777777
126     777777777777777777777777777777777777776771174751647146316461371377777777
127     777777777777777777777777777777777777777370576354763446117475371377777777
128     777777777777777777777777777777777777777570377556273242416677170477777777
129     777777777777777777777777777777777777777677707657271253707664172677777777
130     777777777777777777777777777777777777777737647717074213723437172677777777
131     777777777777777777777777777777777777777757627707476311732506172077777777
132     777777777777777777777777777777777777777767617700016315730157066777777777
133     777777777777777777777777777777777777771771774147637356337563373377777777
134     777777777777777777777777777777777777774774772143617147317563373377777777
135     777777777777777777777777777777777777776376371151647146337461371777777777
136     777777777777777777777777777777777777777377370554763446117475371377777777
137     777777777777777777777777777777777777777577570356273242417677170477777777
138     777777777777777777777777777737777777777677677717271253706264172777777777
139     777777777777777777777777777737777777777077076470742117200160723777777777
140     777777777777777777777777777777377777000000000201777547301470730777777777
141     777777777777777777777777777777377777717417417101777543301430731777777777
142     777777777777777777777777777777377777774764764441777453121510713777777777
143     777777777777777777777777777777377777763663663421775651502550753777777777
144     777777777777777777777777777777377777771671671411777614502570746777777777
145     777777777777777777777777777777377777774274274005777716452154766777777777
146     777777777777777777777777777777377777777077077003777717603164761777777777
147     777777777777777777777777777777777777704004004377003367341472730777777777
148     777777777777777777777777777777777777717617617776403363341433721777777777
149     777777777777777777777777777777777777747647647776201373151512717777777777
150     777777777777777777777777777777777777763763763776105771576753753777777777
151     777777777777777777777777777777777777707707707776044774476745744777777777
152     777777777777777777777777777777777777776376376376026777636365767777777777
153     777777777777777777777777777777777777770770773760163777031747657777777777
154     777777777777777437777777777777776377704004004376174363365633730777777777
155     777777777777777437777777777777776377743743743757432371574713713777777777
156     777777777777777437777777777777776377703703703777013744363457447777777777
157     777777777777777437777777777777776377770770773761607777031747657777777777
158     777777777777777437777777777777777477700000000037777603165633730777777777
159     777777777777777437777777777777777477741741741707777511564711712777777777
160     777777777777777437777777777777777477774174174361777454426345766777777777
161     777777777777777437777777777777777477774774770017774377031647617777777777
```

OUTPUT

Figure 3 (cont.)

```
162   77637777777777777777777777777777777777777777770170172147637357077563373277777777
163   77637777777777777777777777777777777777777777774374370361707146476671371377777777
164   77637777777777777777777777777777777777777777777177177174763462276675174677777777
165   77637777777777777777777777777777777777777777777577577576373672177777576677777777
166   77777777777777777777777777777377777777777777770000000003617357746563373077777777
167   77777777777777777777777777777377777777777777774074075070343443726471170077777777
168   77777777777777777777777777777377777777777777777637637637171673717634576577777777
169   77777777777777777777777777777777607777777777777077077005746317743164763777777777
170   77777777777777777777777777777777607777777777777677677040167357733576767777777777
171   77777777777777777777777777777777777377777777772542547600436301113104172077777777
172   77777777777777777777777777777777777377777777777677676000167357707574767777777777
173   77777777777777777777777777777777777777777777770300300303617146716617373177777777
174   77777777777777777777777777777777777777777777776026036034361463147517174377777777
175   77777777777777777777777777777777777777777777777707717737174631762457576177777777
176   77777777777777777777777777777777777777777777777747747700016735770037476577777777
177   77777777777777777777777777777777777777777777770650650647637356736770373377777777
178   77777777777777777777777777777777777777777777770750770763717146317764371377777777
179   77777777777777777777777777777777777777777777776256276271343463146762174777777777
180   77777777777777777777777777777777777777777777777657657637571671762761576377777777
181   77777777777777777777777417777777777777777777610000000000607146116461407477777777
182   77777777777777777777777417777777777777777777617407407400170621442214207477777777
183   77777777777777777777777417777777777777777777617767767700056735771357107777777777
184   77777777777777777777777407777777777777777777607777777777777777777777047377777777
185   77777777777777777777777407777777777777777777607777777777777777777777027377777777
186   77777777777777777777777407777777777777777777607777777777777777777777017377777777
187   77777777777777777777740377777777777777777770170010010003617146316563774477777777
188   77777777777777777777740377777777777777777770176176176170743467157671772477777777
189   77777777777777777777740377777777777777777770177407407436130631440214771477777777
190   77777777777777777777777767777777777777777777762352351301427144312422707577777777
191   77777777777777777777777777777377777777777777773553557547432215706517170277777777
192   77777777777777777777777777777777777777777777777577407556273242416777170577777777
```

OUTPUT

Figure 3 (cont.)

```
                                   PRIORITY  LIST
   COURSE        CONFLICT        DAYS PER      UNITS PER      UNITS PER        BUFFER
  SECTIONS        COUNTER           CYCLE            DAY   DAY CONSTANT   REQUIREMENT

       181            441               5              1              1             0
       182            441               5              1              1             0
       114            347               5              1              1             0
        63            284               5              1              1             0
         6            263               5              1              1             0
        69            257               5              1              1             0
        64            256               5              1              1             0
        67            254               5              1              1             0
        72            254               5              1              1             0
        12            252               5              1              1             0
        70            239               5              1              1             0
        65            238               5              1              1             0
         5            236               5              1              1             0
        71            236               5              1              1             0
         7            230               5              1              1             0
         1            229               5              1              1             0
        66            229               5              1              1             0
         4            227               5              1              1             0
        11            225               5              1              1             0
        10            221               5              1              1             0
       133            214               5              1              1             0
       171            214               5              1              1             0
       176            214               5              1              1             0
         8            212               5              1              1             0
         2            211               5              1              1             0
         3            209               5              1              1             0
        68            209               5              1              1             0
       187            209               5              1              1             0
       188            209               5              1              1             0
       139            207               5              1              1             0
         9            203               5              1              1             0
        73            202               5              1              1             0
       162            201               5              1              1             0
       163            201               5              1              1             0
       164            201               5              1              1             0
       169            201               5              1              1             0
       115            200               5              1              1             0
       116            200               5              1              1             0
       117            200               5              1              1             0
       124            200               5              1              1             0
       125            200               5              1              1             0
       126            200               5              1              1             0
       134            200               5              1              1             0
       166            200               5              1              1             0
       167            200               5              1              1             0
       168            200               5              1              1             0
       173            200               5              1              1             0
       174            200               5              1              1             0
       175            200               5              1              1             0
       177            200               5              1              1             0
       178            200               5              1              1             0
       179            200               5              1              1             0
       189            195               5              1              1             0
       121            193               5              1              1             0
       130            193               5              1              1             0
```

OUTPUT

Figure 3 (cont.)

```
123   186   5   1   1   0
132   186   5   1   1   0
118   179   5   1   1   0
119   179   5   1   1   0
127   179   5   1   1   0
128   179   5   1   1   0
120   172   5   1   1   0
129   172   5   1   1   0
135   172   5   1   1   0
136   172   5   1   1   0
138   172   5   1   1   0
192   172   5   1   1   0
137   165   5   1   1   0
122   151   5   1   1   0
131   151   5   1   1   0
172   144   5   1   1   0
190   124   5   1   1   0
191   123   5   1   1   0
183   112   5   1   1   0
170   103   5   1   1   0
180    74   5   1   1   0
165    40   5   1   1   0
184    17   5   1   1   0
185    17   5   1   1   0
186    17   5   1   1   0
 30   267   4   1   1   0
 84   266   4   1   1   0
 24   263   4   1   1   0
 18   258   4   1   1   0
 79   259   4   1   1   0
 74   257   4   1   1   0
 77   249   4   1   1   0
 82   249   4   1   1   0
 87   248   4   1   1   0
 75   240   4   1   1   0
 80   240   4   1   1   0
 23   236   4   1   1   0
 19   231   4   1   1   0
 25   231   4   1   1   0
 76   231   4   1   1   0
 81   231   4   1   1   0
 13   230   4   1   1   0
 17   230   4   1   1   0
 85   230   4   1   1   0
 86   230   4   1   1   0
 22   222   4   1   1   0
 28   222   4   1   1   0
 29   222   4   1   1   0
 16   221   4   1   1   0
 20   213   4   1   1   0
 26   213   4   1   1   0
 14   212   4   1   1   0
 21   209   4   1   1   0
 27   204   4   1   1   0
 78   204   4   1   1   0
 83   204   4   1   1   0
 15   203   4   1   1   0
 88   202   4   1   1   0
 59   561   2   1   1   0
111   561   2   1   1   0
```

OUTPUT

Figure 3 (cont.)

```
 60    525    2    1    1    0
112    291    2    1    1    0
 53    265    2    1    1    0
 54    265    2    1    1    0
 55    265    2    1    1    0
 56    265    2    1    1    0
 57    265    2    1    1    0
106    265    2    1    1    0
107    265    2    1    1    0
108    265    2    1    1    0
109    265    2    1    1    0
 36    263    2    1    1    0
 89    263    2    1    1    0
 92    254    2    1    1    0
 90    245    2    1    1    0
 31    236    2    1    1    0
 35    236    2    1    1    0
 91    236    2    1    1    0
 34    227    2    1    1    0
 32    218    2    1    1    0
 33    209    2    1    1    0
 93    209    2    1    1    0
 97    177    2    2    2    0
 96    161    2    2    2    0
110    157    2    1    1    0
 94    152    2    2    2    0
 95    152    2    2    2    0
101    152    2    2    2    0
 98    143    2    2    2    0
 99    143    2    2    2    0
100    143    2    2    2    0
140    139    2    2    2    0
141    139    2    2    2    0
143    139    2    2    2    0
144    139    2    2    2    0
146    139    2    2    2    0
142    132    2    2    2    0
145    132    2    2    2    0
147    118    2    2    2    0
148    118    2    2    2    0
149    118    2    2    2    0
150    111    2    2    2    0
151    111    2    2    2    0
152    111    2    2    2    0
153    111    2    2    2    0
 58     76    2    1    1    0
 61     21    2    1    1    0
 62     21    2    1    1    0
113     21    2    1    1    0
104    314    1    2    2    0
105    305    1    2    2    0
 47    296    1    2    2    0
 48    296    1    2    2    0
 50    296    1    2    2    0
 51    296    1    2    2    0
102    296    1    2    2    0
103    269    1    2    2    0
158    239    1    2    2    0
154    232    1    2    2    0
155    232    1    2    2    0
```

OUTPUT

Figure 3 (cont.)

```
156       232         1         2         2         0
159       232         1         2         2         0
160       232         1         2         2         0
161       190         1         2         2         0
 37       143         1         2         2         0
 38       143         1         2         2         0
 39       143         1         2         2         0
 40       143         1         2         2         0
 42       143         1         2         2         0
 43       143         1         2         2         0
 44       143         1         2         2         0
 45       143         1         2         2         0
 41       134         1         2         2         0
 46       125         1         2         2         0
157        99         1         2         2         0
 52        98         1         2         2         0
 49        89         1         2         2         0
```

```
                                         MASTER SCHEDULE DAY      1

PERIOD

   1      181    114      6     70      1     11     10    176    187    168     76     26     21
   2      182     63     69     67     65    133    188      9     73    170     81     22     20
   3       64     72     12      7    171    162    134    174    183     75     17     86     28     83     32
   4        5     66      8    139    163    115    167    184     34     18     77     25     16     33    110
   5       71      2    164    169    116    124    120    135    185 [illegible] 79     19     15     88     56     92
   6        4    117    125    175    132    136    138    165    186 [illegible] 74     82     85     29     54     91
   7        3    126    189    118     87     80     13     78     36    141     61     40
   8       40    141     68    179    121    119     14     27    107     31     62
```

OUTPUT

Figure 3 (cont.)

OUTPUT

Figure 3 (cont.)

```
                                          MASTER SCHEDULE DAY        2
PERIOD

     1        181   114     6    70     1    11    10   176   187   168    76    26    21
     2        182    63    69    67    65   133   188     9    73   170    81    22    20
     3         64    72    12     7   171   162   134   174   183    75    17    86    28    83    32
     4          5    66     8   139   163   115   167   184    84    18    77    25    16    33   110
     5         71     2   164   169   116   124   120   135   185    30    79    19    15    88    56    92
     6          4   117   125   175   132   136   138   165   186    24    74    82    85    29    54    91
     7          3   126   189   118    87    60    13    78    36    61
     8         68   166   179   172    23    14    27   107    31    62
```

OUTPUT

Figure 3 (cont.)

```
                                        MASTER SCHEDULE DAY        3
PERIOD

     1      181  114    6   70    1   11   10  176  187  168   76   26   21
     2      182   63   69   67   65  133  168    9   73  170   81   22   20
     3       64   72   12    7  171  162  134  174  183   75   17   86   28   83   38
     4       38    5   66  139  163  115  167  164   84   24   82   25   16   93
     5       71    8  164  169  116  124  120  135  185   30   79   77   13   88   55   70
     6        4    2  117  125  175  132  136  138  165  186   18   74   87   19   29
     7        3  126  189  118   80   14        89  100  141  113   41   49
     8       41   49  100  141   68  179   12  119   85   27   34
```

OUTPUT

Figure 3 (cont.)

```
                                        MASTER SCHEDULE DAY      4
PERIOD

     1     181   114     6    70     1    11    10   176   187   168    76    26    21
     2     182    63    69    67    65   133   188     9    73   170    81    22    20
     3      64    72    12     7   171   162   134   174   183    75    17    86    28    83
     4       5    66     8   139   163   115   167   184    84    18    77    25    16    93
     5      71     2   164   169   116   124   120   135   185    30    79    19    15    88    90
     6       4   117   125   175   132   136   138   165   186    24    74    82    85    29    43
     7      43     3   126   189   118    67    80    78    89   113    52
     8      52    68   166   179   172    13    27   108    34
```

OUTPUT

Figure 3 (cont.)

```
                                        MASTER SCHEDULE DAY        5

PERIOD

    1       181   114     6    70     1    11    10   176   187   168    14
    2       182    53    69    67    65   133   188     9    73   170
    3        64    72    12     7   171   162   134   174   183    15    35
    4         5    66     8   139   163   115   167   184    55    97   101
    5        97   101    71     2   164   169   116   124   120   135   185    23
    6         4   117   125   175   132   136   138   165   186    57    96    98   103
    7        96    98   103     3   126   169   118
    8        68   166   179   172    59   108
```

```
                                                          MASTER SCHEDULE DAY        6
PERIOD

      1        166    172    180     23    111    109    150    159     39     44
      2         39     44    150    159    177    121    123    191    112     53    102
      3        102    173    178    119    129     59     57     97    101    153    161
      4         97    101    153    161    127    192     60    106     47
      5         47    130    128    122     96     98    143    148     58     50
      6         50     96     98    143    148    137    131     48
      7         48    190     94     99    144    149     51
      8         51     94     99    144    149
```

OUTPUT

Figure 3 (cont.)

```
                                                 MASTER SCHEDULE DAY        7
PERIOD

     1       166    172    180     23    111    109    150     46
     2        46    150    173    177    119     60     53     94    143    157
     3        94    143    157    178    128    129     35     99
     4        99    121    123    127    192    122    112    106     37
     5        37    130    137    190    100    149     58    104
     6       100    104    149    131    144    156
     7       144    156     95    147    105     42
     8        42     95    105    147
```

OUTPUT

Figure 3 (cont.)

OUTPUT

Figure 3 (cont.)

```
                                   MASTER SCHEDULE DAY        8
PERIOD

     1      173   177   119   146   152    45
     2       45   146   152   178   127   192    95
     3       95   121   123   128   127   122   155
     4      155   130   137   180   142   148
     5      142   148   131
     6      190   145   151
     7      145   151   158   154
     8      154   158
```

OUTPUT

Figure 3 (cont.)

```
                              MASTER SCHEDULE DAY        9
PERIOD

     1       173   177   128   146   152
     2       146   152   178   127   192
     3       130   123   137   122
     4       129   190   151
     5       153   180   140
     6       140   147   160
     7       147   160   142
     8       142   131
```

```
                                   MASTER SCHEDULE DAY        10

PERIOD

     1       173    177    128
     2       178    130    123    127    137
     3       129    122
     4       192    190
     5       131    145    151
     6       145    151
     7       191    180
```

OUTPUT

Figure 3 (cont.)

```
                    MASTER SCHEDULE DAY      11

PERIOD

     1      191
     2      140
     3      140
```

OUTPUT

Figure 3 (cont.)

```
                                   MASTER SCHEDULE DAY       12
PERIOD                                                        .
   1      191
```

OUTPUT

Figure 3 (cont.)

```
                                          MASTER SCHEDULE DAY       13
      PERIOD
          1       191
```

OUTPUT

Figure 3 (cont.)

```
NUMBER OF DAYS=    15     PERIOD=     2
0000000000  000000000000000000000000000000000000000000000000000000000000000
  1961
```

OUTPUT

Figure 3 (cont.)

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4

```
Z                 100015                DECK
 TURKSEN                  RINDUE 00395                045   100   000
$EXECUTE,DUMP
$ASSEMBLE,PUNCH OBJECT
*00943      TURKSEN      PITT
*       XEQ
*       LABEL
*        FAP
         ENTRY   IZERO
IZERO    SXA     X2,2
         CLA     1,4
         STA     CR
         CLA*    2,4
         PDX     **,2
         TXI     *+1,2,-1
         SXD     TEXT,2
         AXT     0,2
CR       STZ     **,2
         TXI     *+1,2,1
TEXT     TXL     *-2,2,**
X2       AXT     **,2
         TRA     3,4
         END
$ASSEMBLE,PUNCH OBJECT
*       LABEL
*        FAP
         ENTRY   IONES
IONES    SXA     X2,2
         CLA     1,4
         STA     CR
         CLA*    2,4
         PDX     **,2
         TXI     *+1,2,-1
         SXD     TEXT,2
         AXT     0,2
         ZAC
         COM
CR       SLW     **,2
         TXI     *+1,2,1
TEXT     TXL     *-2,2,**
X2       AXT     **,2
         TRA     3,4
         END
$ASSEMBLE,PUNCH OBJECT
*       LABEL
*        FAP
         ENTRY   IPSIG
IPSIG    STZ*    2,4
         CAL*    1,4
         TZE     *+2
         STL*    2,4
         TRA     3,4
         END
$ASSEMBLE,PUNCH OBJECT
*       LABEL
*        FAP
         ENTRY   SCONS
.SCONS   SXA     XR2,2
         CLA     1,4
         STA     X
```

```
        CLA     2,4
        STA     Y
        CLA     3,4
        STA     Z
        LDQ     =0777777777776
        STQ     XSET
        LDQ     =1
        STQ     YSET
        STZ     ZSET
        AXT     0,2
        CAL     =0777777777777
LOOP    LDQ     XSET
        ROL     35
        STQ     XSET
X       STQ     **,2
        LDQ     YSET
        ROL     35
        STQ     YSET
Y       STQ     **,2
        LDQ     ZSET
        LGR     1
        STQ     ZSET
Z       STQ     **,2
        TXI     *+1,2,1
        TXL     LOOP,2,35
XR2     AXT     **,2
        TRA     4,4
XSET    PZE
YSET    PZE
ZSET    PZE
        END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RTRM(K,KK,MM)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
      DO 1054 J=K,KK
 1054 READ TAPE MRTAPE,(R(L),L=1,MM)
      RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
       SUBROUTINE LOCI(ISNI,I,NCSS)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
      DO 2145 L=I,NCSS
       IF( ISNI-ISN(L))2145,2146,2145
 2145 CONTINUE
 2146 I=L
      RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
*     LABEL
      SUBROUTINE RVIB(I,MRW,MRP,MM,IS)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
      DO 1800 J=1,300
 1800 ISYM(J)=0
      IS=1
      ISYM(1)=ISN(I)
 1801 IT=0
      IF(I-NRTO)1806,1806,1808
 1806 REWIND MRT2
      KG=1
      CALL RRMO(KG,I,MM)
      GO TO 1819
 1808 REWIND MRT1
      IT=1
      KG=1
      I5=I-NRTO
      CALL RRMT (KG,I5,MM)
 1819 DO 1820 L=1,MM
B      RE(L)= R(L)* WV(L)
B      RE(L)= RE(L)* DV(L)
B1820  RE(L)= RE(L)* AV(L)
      IUPD(I)=IUPD(I)-1
      IF(IUPD(I))1826,1830,1840
 1826 ISTOP=1626
      WRITE OUTPUT TAPE IOTAPE,1827,ISTOP
 1827 FORMAT(1H ,I6)
      CALL ERROR
 1830 CALL UPDZ(I,MRW,MRP)
      GO TO 1850
B1840  CV1(MRW)= CV1(MRW)+ Y(MRP)
      IFNP=1
 1850 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE UPDZ(I,MRW,MRP)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
B      DV(MRW)= DV(MRW)* X(MRP)
      IDPW(I)=IDPW(I)-1
      IF(IDPW(I))1855,1860,1870
 1855 ISTOP=1855
      WRITE OUTPUT TAPE IOTAPE,1856,ISTOP
 1856 FORMAT(1H ,I6)
      CALL ERROR
B1860  WV(MRW)= WV(MRW)* X(MRP)
      GOTO 1880
 1870 IUPD(I)=IUPDC(I)
 1880 RETURN
      END
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
$COMPILE FORTRAN,PUNCH OBJECT
*      LABEL
       SUBROUTINE JTHC(I,MRW,MRP,NRPS,MM,IS)
     1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
     11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
     12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
       DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
      100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
      2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
B       PC= RE(MRW)* Y(MRP)
       CALL IPSIG(PC,FLAG)
       IF(FLAG)1950,1940,1950
 1940 GO TO 2060
 1950 IF(IUPD(I)-NRPS)1960,1960,1980
 1960 IF(IDB(I))1935,2000,1990
 1935 ISTOP=1935
       WRITE OUTPUT TAPE IOTAPE,1936,ISTOP
 1936 FORMAT(1H ,I6)
       CALL ERROR
B1980  AV(MRW)= AV(MRW)* X(MRP)
       GO TO 2060
B1990  PC= DB1(MRW)* Y(MRP)
       CALL IPSIG(PC,FLAG)
       IF(FLAG)1995,2060,1995
B1995  DB2(MRW)= DB2(MRW)* X(MRP)
 2000 IS=IS+1
       ISYM(IS)=ISN(I)
 2001 IF(IT)1935,2002,2010
 2002 IF(I-NRTO)2005,2005,2004
 2005 CALL RRMO(KG,I,MM)
       GO TO 2019
 2004 REWIND MRT1
       IT=1
       KG=1
       I5=I-NRTO
       CALL RRMT(KG,I5,MM)
       GO TO 2019
 2010 I5=I-NRTO
       CALL RRMT(KG,I5,MM)
 2019 DO 2020 L=1,MM
B2020  RE(L)= RE(L)* R(L)
       IUPD(I)=IUPD(I)-1
       IF(IUPD(I))1935,2040,2050
 2040 CALL UPDZ(I,MRW,MRP)
       GO TO 2060
B2050  CV1(MRW)= CV1(MRW)+ Y(MRP)
       IFNP=1
 2060 RETURN
       END
$COMPILE FORTRAN,PUNCH OBJECT
*      LABEL
       SUBROUTINE RRMO(KG,I,MM)
     1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
     11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
     12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
       DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
      100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
      2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
       DO 1810 J=KG,I
 1810 READ TAPE MRT1,(R(L),L=1,MM)
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
      IF(I-NRTO)4001,4002,4002
 4002 REWIND MRT1
      KG=1
       GO TO 4005
 4001 KG=I+1
 4005 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
      SUBROUTINE RRMT(KG,I5,MM)
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
      DO 1815 J=KG,I5
 1815 READ TAPE MRT2,(R(L),L=1,MM)
      IF(I-NCSS)4001,4002,4002
 4002 REWIND MRT2
      KG=1
      IT=0
      GO TO 4005
 4001 KG=I5+1
 4005 RETURN
      END
$COMPILE FORTRAN,PUNCH OBJECT
*     LABEL
C            MAIN PROGRAM
    1 COMMON X,Y,Z,ISN,ICC,IDPW,IUPD,IUPDC,IDB,WV,DV,AV,CV1,CV2,DB1,DB2,
    11R,RE,ISYM,IRBS,ISP,IOTAPE,MRTAPE,MRT1,MRT2,NRTO,IFNP,
    12K,KK,MM,KG,I,I5,ISNI,NCSS,MRP,MRW,IS,NRPS,ISC,IJ,NP,IT
      DIMENSION X(36),Y(36),Z(36),ISN(1000),ICC(1000),IDPW(1000),IUPD(10
     100),IUPDC(1000),IDB(1000),WV(100),DV(100),AV(100),CV1(100),CV2(100
     2),DB1(100),DB2(100),R(100),RE(100),ISYM(300),IRBS(14),ISP(4)
      IRTAPE=3
      ISTAPE=2
      MRTAPE=9
      ITAPE=7
      IOTAPE=6
      MRT2=2
      MRT1=3
      CALL SCONS(X,Y,Z)
      READ INPUT TAPE  ITAPE,15,NCSS,IPD,IPM,IPA,NRB
   15 FORMAT(5I6)
      REWIND ISTAPE
      REWIND IRTAPE
      MM=NCSS/36
      MMP=NCSS-MM*36
      IF(MMP)90,16,18
   16 MMP=36
      MMM=MM
      GO TO 19
   18 MMM=MM
      MM=MM+1
      NBO=NRB/2
      NBT=NRB-NBO
   19 DO 50 J=1,NBO
      READ INPUT TAPE  ITAPE,51,(IRBS(L),L=1,14)
   51 FORMAT(14I5)
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
 50 WRITE TAPE IRTAPE,(IRBS(L),L=1,14)
    END FILE IRTAPE
    REWIND IRTAPE
    DO 12 J=1,NBT
    READ INPUT TAPE  ITAPE,51,(IRBS(L),L=1,14)
 12 WRITE TAPF ISTAPE,(IRBS(L),L=1,14)
    END FILE ISTAPE
    REWIND ISTAPE
    DO 20 J=1,NCSS
 20 ICC(J)=0
    REWIND MRTAPE
    WRITE OUTPUT TAPE IOTAPE,35
 35 FORMAT(1H1/52X,16HRELATIONS MATRIX//50X,18HELEMENTS ARE OCTAL//55X
 351,8HDECOD AS//57X,5H0=000/57X,5H1=001/57X,5H2=010/57X,5H3=011/57X,5
 352H4=100/57X,5H5=101/57X,5H6=110/57X,5H7=111//24X,3HROW//)
 21 DO 300 K=1,NCSS
 30 REWIND IRTAPE
    REWIND ISTAPE
    CALL IONES (R,MM)
    DO 200 J=1,NRB
    IF(J-NBO)9,9,8
  9 REWIND ISTAPE
    READ TAPE IRTAPE,(IRBS(L),L=1,14)
    GO TO 5
  8 REWIND IRTAPE
    READ TAPE ISTAPE,(IRBS(L),L=1,14)
  5 DO 70 L=1,14
    IF(K-IRBS(L))70,60,70
 70 CONTINUE
    GO TO 200
 60 IRBS(L)=0
    DO 80 L=1,14
    IF(IRBS(L))90,80,100
 90 ISTOP=90
 91 WRITE OUTPUT TAPE IOTAPE,95
 95 FORMAT(1H0,35HDATA IS BAD.LOOK AT R-ARRAY IN DUMP)
    GO TO 5000
100 ICC(K)=ICC(K)+1
    MRW=IRBS(L)/36
    MRP=IRBS(L)-MRW*36
    IF(MRP)90,120,130
120 MRP=36
    GO TO 135
130 MRW=MRW+1
135 IF(MRW-100)140,140,150
150 ISTOP=150
    GO TO 91
B 140  R(MRW)= R(MRW)* X(MRP)
 80 CONTINUE
200 CONTINUE
    WRITE TAPE MRTAPE,(R(L),L=1,MM)
    WRITE OUTPUT TAPE IOTAPE,210,K,(R(L),L=1,MM)
210 FORMAT(1H ,20X,I6,3X,7O12)
300 CONTINUE
    REWIND IRTAPE
    REWIND ISTAPE
    END FILE MRTAPE
    REWIND MRTAPE
    REWIND ISTAPE
    DO 53 J=1,NCSS
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
      READ INPUT TAPE ITAPE,52,ISN(1),IDPW(1),IUPD(1),IDB(1)
   52 FORMAT(4I6)
   53 WRITE TAPE ISTAPE,ISN(1),IDPW(1),IUPD(1),IDB(1)
      END FILE ISTAPE
      REWIND ISTAPE
      DO 310 K=1,NCSS
  310 ISN(K)=K
      DO 400 K=1,NCSS
      DO 500 L=2,NCSS
      IF(ICC(L)-ICC(L-1))500,500,450
  450 ITEM=ICC(L-1)
      ITEN=ISN(L-1)
      ICC(L-1)=ICC(L)
      ISN(L-1)=ISN(L)
      ICC(L)=ITEM
      ISN(L)=ITEN
  500 CONTINUE
  400 CONTINUE
      REWIND ISTAPE
      DO 900 K=1,NCSS
      READ TAPE ISTAPE,(ISP(J),J=1,4)
      DO 800 I=1,NCSS
      IF(ISN(I)-ISP(1))800,850,800
  850 IDPW(I)=ISP(2)
      IUPD(I)=ISP(3)
      IUPDC(I)=ISP(3)
      IDB(I)=ISP(4)
      GO TO 900
  800 CONTINUE
      WRITE OUTPUT TAPE IOTAPE,860,(ISP(J),J=1,4)
  860 FORMAT(1H0,15HBAD INFORMATION,4I5)
  900 CONTINUE
      REWIND ISTAPE
      K=5
      WRITE OUTPUT TAPE IOTAPE,941
  941 FORMAT(1H1,53X,14HPRIORITY  LIST)
      WRITE OUTPUT TAPE IOTAPE,940
  940 FORMAT(1H0,9X,6HCOURSE,7X,8HCONFLICT,7X,8HDAYS PER,6X,9HUNITS PER,
 94016X,9HUNITS PER,9X,6HBUFFER/8X,8HSECTIONS,8X,7HCOUNTER,10X,5HCYCLE,
 940212X,3HDAY,15H   DAY CONSTANT,15H    REQUIREMENT/)
  950 DO 1000 I=1,NCSS
      IF(K-IDPW(I))1000,1010,1000
 1010 WRITE TAPE ISTAPE,ISN(I),ICC(I),IDPW(I),IUPD(I),IUPDC(I),IDB(I)
      WRITE OUTPUT TAPE IOTAPE,1005,ISN(I),ICC(I),IDPW(I),IUPD(I),
     1IUPDC(I),IDB(I)
 1005 FORMAT(1H ,6I15)
 1000 CONTINUE
      K=K-1
      IF(K)90,1030,950
 1030 END FILE ISTAPE
      REWIND ISTAPE
      DO 1050 I=1,NCSS
 1050 READ  TAPE ISTAPE,ISN(I),ICC(I),IDPW(I),IUPD(I),IUPDC(I),IDB(I)
      REWIND ISTAPE
 1055 NRTO=NCSS/2
      REWIND MRTAPE
      REWIND MRT1
      REWIND MRT2
      KK=ISN(1)
      K=1
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
      CALL RTRM(K,KK,MM)
      WRITE TAPE MRT1,(R(L),L=1,MM)
      DO 1080 I=2,NCSS
      IF(ISN(I)-ISN(I-1))1070,1069,1060
 1069 ISTOP=1069
      GO TO 5000
 1070 REWIND MRTAPE
      K=1
      GO TO 1063
 1060 K=KK+1
 1063 KK=ISN(I)
 1065 CALL RTRM(K,KK,MM)
      IF(I-NRTO)1066,1066,1068
 1066 WRITE TAPE MRT1,(R(L),L=1,MM)
      GO TO 1080
 1068 WRITE TAPE MRT2,(R(L),L=1,MM)
 1080 CONTINUE
      END FILE MRT1
      END FILE MRT2
      REWIND MRTAPE
      REWIND MRT1
 1085 REWIND MRT2
      WRITE OUTPUT TAPE IOTAPE,5010
 5010 FORMAT(1H1,52X,15HMASTER SCHEDULE)
      CALL IONES(WV,MM)
      CALL IONES(DB2,MM)
      IF(MMP-36)1152,1153,1390
B1152  WV(MM)= WV(MM)* Z(MMP)
 1153 CALL IZERO(CV1,MM)
      CALL IZERO(CV2,MM)
      ND=0
 1545 ND=ND+1
 1690 WRITE OUTPUT TAPE IOTAPE,1700 ,ND
 1700 FORMAT(1H1,46X,22HMASTER SCHEDULE DAY    ,I6//6HPERIOD/)
      CALL IONES(DV,MM)
      CALL IONES(AV,MM)
      DO 1550 J=1,MM
 1550 DB1(J)=DB2(J)
      CALL IONES(DB2,MM)
 1160 IPS=IPM
      NP=1
      NPS=0
      NRPS=IPS
      IMA=1
      IAOS=0
      IFNP=0
 1309 REWIND MRT1
      REWIND MRT2
      IF(IFNP)1310,1320,2100
 1310 ISTOP=1310
      GO TO 5000
 1320 REWIND MRT1
      REWIND MRT2
      DO 1560 I=1,NCSS
 1321 IF(ISN(I))1310,1310,1340
 1340 MRW=ISN(I)/36
      MRP=ISN(I)-MRW*36
      IF(MRP)1310,1360,1370
 1360 MRP=36
      GO TO 1375
 1370 MRW=MRW+1
 1375 IF(MRW-MM)1380,1380,1390
 1390 IST$P=1390
      GO TO 5000
B1380  PC= WV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1420,1560,1420
B1420  PC= DV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1430,1560,1430
B1430  PC= AV(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1440,1750,1440
 1750 IAOS=1
      GO TO 1560
 1440 IF(IUPD(I)-NRPS)1450,1450,1520
B1520  AV(MRW)= AV(MRW)* X(MRP)
      GO TO 1750
 1450 IF(IDB(I))1310,1470,1510
B1510  PC= DB1(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1512,1560,1512
B1512  DB2(MRW)= DB2(MRW)* X(MRP)
      GO TO 1470
 1560 CONTINUE
      DO 1564 J=1,MM
      PC=WV(J)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1565,1564,1565
 1564 CONTINUE
 1610 WRITE OUTPUT TAPE IOTAPE,1620,ND,NP
 1620 FORMAT(1H1,15HNUMBER OF DAYS=,I6,5X,7HPERIOD=,I6)
      WRITE OUTPUT TAPE IOTAPE,1640,(WV(L),L=1,MM)
 1640 FORMAT(1H0,10012)
 1641 ISTOP=1641
      WRITE OUTPUT TAPE IOTAPE,1642,ISTOP
 1642 FORMAT(1H0,I6)
      CALL SYSTEM
 1565 IF(IMA)1310,1545,1570
 1570 IMA=0
      IF(IAOS)1310,1545,1650
 1650 WRITE OUTPUT TAPE IOTAPE,1660
 1660 FORMAT(1H0,51X,18H AFTERNOON SESSION)
 1720 IPS=IPA
      IMA=0
      IF(IPS)1310,1545,1730
 1730 NPS=0
      NRPS=IPS
      CALL IONES(AV,MM)
      GO TO 1320
 1470 CALL RVIB(I,MRW,MRP,MM,IS)
      II=I+1
 1475 DO 1480 I=II,NCSS
      MRW=ISN(I)/36
      MRP=ISN(I)-MRW*36
      IF(MRP)1310,1890,1900
 1890 MRP=36
      GO TO 1905
 1900 MRW=MRW+1
 1905 IF(MRW-MM)1910,1910,1390
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
B1910  PC= CV2(MRW)* Y(MRP)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)1480,1930,1480
 1930 CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 1480 CONTINUE
 1490 WRITE OUTPUT TAPE IOTAPE,1530,NP,(ISYM(J),J=1,IS)
 1530 FORMAT(1H ,I5,3X,18I6/(9X,18I6))
      NP=NP+1
      IF(NP-IPD)1710,1710,1545
 1710 NPS=NPS+1
      NRPS=IPS-NPS
      IF(NRPS)1310,1650,1309
 2100 IFNP=0
      DO 2110 J=1,MM
 2110 CV2(J)=CV1(J)
      CALL IZERO(CV1,MM)
      DO 2130 J=1,MM
      DO 2130 K=1,36
B      PC= CV2(J)* Y(K)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2140,2130,2140
 2130 CONTINUE
 2135 ISTOP=62135
 5000 WRITE OUTPUT TAPE IOTAPE,2136,ISTOP
 2136 FORMAT(1H ,I6)
      CALL ERROR
 2140 MRW=J
      MRP=K
      ISNI=(MRW-1)*36+MRP
      I=1
      CALL LOCI(ISNI,I,NCSS)
      CALL RVIB(I,MRW,MRP,MM,IS)
      IF(K-36)2150,2180,2180
 2150 K=K+1
      DO 2170 LL=K,36
B      PC= CV2(J)* Y(LL)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2160,2170,2160
 2160 MRW=J
      MRP=LL
      ISNI=(MRW-1)*36+MRP
       I=1
      CALL LOCI(ISNI,I,NCSS)
      KG=1
       IT=0
      REWIND MRT1
      REWIND MRT2
      CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 2170 CONTINUE
       IF(MM-J)1310,2231,2180
 2180 JJ=J+1
       DO 2200 J=JJ,MM
       DO 2200 K=1,36
B     PC=CV2(J)*Y(K)
      CALL IPSIG(PC,FLAG)
      IF(FLAG)2210,2200,2210
 2210 MRW=J
      MRP=K
      ISNI=(MRW-1)*36+MRP
       I=1
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
      CALL LOCI(ISNI,I,NCSS)
      KG=1
       IT=0
      REWIND MRT1
      REWIND MRT2
      CALL JTHC(I,MRW,MRP,NRPS,MM,IS)
 2200 CCNTINUE
 2231 II=1
      REWIND MRT1
      REWIND MRT2
      IT=0
      KG=1
      GO TO 1475
      END
```

PROGRAM OF THE CONTROLLED SCHEDULE CYCLE

Figure 4 (cont.)

```
                PARAMETER CARDS
192      8       8       0   538       5



                  COURSE REQUEST BLOCK CARDS
  1     7    13    19    25    31    42    47    53    59
  1     7    13    19    25    31    37    50    53    59
  2     8    14    20    26    32    37    50    53    59
116   125   134   154   148   166   178   181
116   125   134   158   141   166   178   181
116   125   134   158   141   166   178   181
115   125   134   154   148   166   173   181
116   125   134   154   149   166   178   181
116   125   134   158   142   166   173   181
116   125   134   154   149   166   173   181
116   125   134   155   149   167   173   181
116   125   134   155   149   167   173   187
116   125   134   155   149   167   178   181
116   125   134   158   142   167   178   181
115   125   134   158   142   167   173   181
116   125   134   155   149   167   178   187
116   125   134   159   142   167   173   187
116   125   134   155   149   167   173   181
116   125   134   155   149   167   178   187
116   125   134   159   142   167   173   187
116   125   134   159   142   167   178   181
116   125   134   155   149   167   173   181
116   125   134   155   149   167   173   181
116   125   134   159   142   163   178   181
117   126   134   159   142   163   178   190
117   126   134   159   142   167   173   181
117   126   134   159   142   163   178   181
117   126   134   155   149   167   173   181
```

SAMPLE INPUT DATA

Figure 5

```
COURSE SECTION SPECIFICATION CARDS
          164     5     1     0
          165     5     1     0
          166     5     1     0
          167     5     1     0
          168     5     1     0
          169     5     1     0
          170     5     1     0
          171     5     1     0
          172     5     1     0
          173     5     1     0
          174     5     1     0
          175     5     1     0
          176     5     1     0
          177     5     1     0
          178     5     1     0
          179     5     1     0
          180     5     1     0
          181     5     1     0
          182     5     1     0
          183     5     1     0
          184     5     1     0
          185     5     1     0
          186     5     1     0
          187     5     1     0
          188     5     1     0
          189     5     1     0
          190     5     1     0
          191     5     1     0
          192     5     1     0
```

SAMPLE INPUT DATA

Figure 5 (cont.)

```
                        RELATIONS MATRIX

                       ELEMENTS ARE OCTAL

                            DECOD AS

                             0=000
                             1=001
                             2=010
                             3=011
                             4=100
                             5=101
                             6=110
                             7=111

ROW

  1     57373737373736355475607777777777777777777777777777777777777777777777777777
  2     37575757575716155475707777777777777777777777777777777777777777777777777777
  3     70676767676706544434771477777777777777637777777777777777777777777777777777
  4     64737373737322444514771477777777777777637777777777777777777777777777777777
  5     62757575747530644140671477777777777777637777777777777777777777777777777777
  6     61767676767675622363671477777777777777777777777777777777777777777777777777
  7     37573734373736355475676377777777777777777777777777777777777777777777777777
  8     57375754575716155475776377777777777777777777777777777777777777777777777777
  9     67736767676706544434777777777777777777777777777777777777776077777777777777777777
 10     73677373737322444514777777777777777777777777777777777777776077777777777777777777
 11     75777575747530644140677777777777777777777777777777777777777777777777777777
 12     76777676767675622363677777777777777777777777777777777777777777777777777777
 13     37374137373736355475677757777777777777777777777777777777777777777777777777
 14     57572157575716155475777757777777777777777777777777777777777777777777777777
 15     57671167676706544434777757777777777777777777777777777777777777777777777777
 16     73730573737322444514777757777777777777777777777777777777777777777777777777
 17     75750375747530644140677757777777777777777777777777777777777777777777777777
 18     76767774757675622363677777017777777777777777777777777777777777777777777777
 19     37373756373736355475677760777777777777777777777777777777777777777777777777
 20     57575735575716155475777760777777777777777777777777777777777777777777777777
 21     67676777570006544434777777777407777777777777777777777777777777777777777777
 22     73737377007322444514777777777777777777777777777777777777777777777777777777
 23     75157476747530644140676577017777777777777777777777777777777777777777777777
 24     76167615767675622363676360777777777777777777777777777777777777777777777777
 25     37373733403736355475677777777777777777777777777777777777777777777777777777
 26     57575753205716155475777777777777777777777777777777777777777777777777777777
 27     67676763106706544434777777777777777777777777777777777777777777777777777777
 28     73737373047322444514777777777777777777777777777777777777777777777777777777
 29     75757571027530644544777777777777777777777777777777777777777777777777777777
 30     74747470017475622363677777777777777777777777777777777777777777777777777777
 31     37373727574036355475677777777407777777777777777777777777777777777777777777
 32     57575747572016155475777777777407777777777777777777777777777777777777777777
 33     57676767671006544434777777777407777777777777777777777777777777777777777777
 34     75737363730422444514777777777407777777777777777777777777777777777777777777
 35     75757565740230644140677777777407777777777777777777777777777777777777777777
 36     76767666760175622363677777777407777777777777777777777777777777777777777777
 37     01010101010141775475677777777770377777777777777777777777777777777777777777
 38     47474747474721775634777777777770377777777777777777777777777777777777777777
 39     63636363636311774716777777777770377777777777777777777777777777777777777777
 40     71717171717105776746777777777770377777777777777777777777777777777777777777
 41     74747474747403776362677777777770377777777777777777777777777777777777777777
```

OUTPUT

Figure 6

OUTPUT

Figure 6 (cont.)

```
 42   010101010101770175756777777777777417777777777777777777777777777777777777777
 43   171717171717764174757777777777777417777777777777777777777777777777777777777
 44   434343434343762076147777777777777417777777777777777777777777777777777777777
 45   707070707070761277466777777777777417777777777777777777777777777777777777777
 46   767676767676760637636777777777777417777777777777777777777777777777777777777
 47   010101010101761634756777777777777763777777777777777037777777777777777777
 48   606060606060774136046777777777777763777777777777777037777777777777777777
 49   767676767676777077636777777777777763777777777777777037777777777777777777
 50   010101010101077760346777777777777774777777777777777417777777777777777777
 51   606060606060617753066777777777777774777777777777777417777777777777777777
 52   747474747674757747636777777777777774777777777777777417777777777777777777
 53   010101010101363556016777777777777777017777777777777777777777777077777777
 54   070707070707171455017777777777777777017777777777777777777777777077777777
 55   636363636363475644407777777777777777017777777777777777777777777077777777
 56   717171717171634664227777777777777777017777777777777777777777777077777777
 57   747474747474716220126777777777777777017777777777777777777777777077777777
 58   747474747674757330076777777777777777017777777777777777777777777077777777
 59   010101010101161454361777777777777777617777777777777777777770077777777777
 60   616161616161414664705177777777777777617777777777777777777770077777777777
 61   777777777777777777774577777777777777617777777777777777777770077777777777
 62   777777777777777777774377777777777777617777777777777777777770077777777777
 63   343434343634346001637777777777777777767777777777777777777777707377777777
 64   177777777777777777777746757367571465166777777777777777777777777777777777
 65   177777777777777777777727567563671465067777777777777777777777777777777777
 66   177777777777777777777717573675730420263777777777777777777777777777777777
 67   607777777777777777777775475736756312636777777777777777777777777777777777
 68   607777777777777777777773436757360140302777777777777777777777777777777777
 69   771777747777777777777737557367571465166777777777777777777777777777777777
 70   771777747777777777777756767563671465067777777777777777777777777777777777
 71   607777777777777777777761673675730420263777777777777777777777777777777777
 72   607777777777777777777771575736756312636777777777777777777777777777777777
 73   777777777777777777777775776757360140302777777777777777777077777775777777
 74   777701777777777777777736777367571465166777777777777777777777777777777777
 75   777777167777777777777757370563671465067777777777777777777777777777777777
 76   777777167777777777777767564675730420263777777777777777777777777777777777
 77   777777167777777777777773662736756312636777777777777777777777777777777777
 78   777777167777777777777775721757360140302777777777777777777777777777777777
 79   777776757777777777777736757407571465166777777777777777777777777777777777
 80   777776757777777777777757367203671465067777777777777777777777777777777777
 81   777776757777777777777767573115730420263777777777777777777777777777777777
 82   777776757777777777777773675056756312636777777777777777777777777777777777
 83   777776757777777777777775736037360140302777777777777777777777777777777777
 84   777777777777777777777716347170471465166777777777777777771777777777777777
 85   777777777777777777777757367564671465067777777777777777771777777777777777
 86   777777777777777777777767573662730420263777777777777777771777777777777777
 87   777777777777777777777773675721756312636777777777777777771777777777777777
 88   777777777777777777777775736757760140302777777637777777777637777777777777
 89   777777677007777777777736757367601465166777777777777777777777777777777777
 90   777777677007777777777757367563501465067777777777777777777777777777777777
 91   777777677007777777777767573675440420263777777777777777777777777777777777
 92   777777677007777777777773675736426312636777777777777777777777777777777777
 93   777777677007777777777775736757010140302777777777777777777777777777777777
 94   77777777777701777777770410204102437+142777777777777777777777777777777777
 95   777777777777017777777704102041022375443777777777777777777777777777777777
 96   777777777777017777777761430614301374612777777777777777777777777777777777
 97   777777777777017777777771634716340774712777777740177777777777777777777777
 98   777777777777603777777704102041027607166777777777777777777777777777777777
 99   777777777777603777777706143061437507467777777777777777777777777777777777
100   777777777777603777777763471634717443622777777777777777777777777777777777
101   777777777777603777777771634716347433712777777777777777777777777777777777
```

OUTPUT

Figure 6 (cont.)

```
102    77777777777777774377770410204102743306677777777777770377777777777777777777
103    77777777777777774377776143061430770760277777777777770377777777777777777777
104    77777777777777774377770410204102037604277777777777777417777777777777777777
105    77777777777777774377776143061430237561277777777777777417777777777777777777
106    77777777777777777740370410204102356540677777777777777777777777770777777777
107    77777777777777777740371634716347146520777777777777777777777777770777777777
108    77777777777777777740374306143061462010277777777777777777777777770777777777
109    77777777777777777740377163471634631205677777777777777777777777770777777777
110    77777777777777777740377573675736035003277777777777777777777777770777777777
111    77777777777777777777410410204102142107177777777777777777777777007777777777
112    77777777777777777777416553265532031064577777777777777777777777007777777777
113    77777777777777777777417777777777777776377777777777777777777777007777777777
114    77777777777777777777776306143061421002177777777777777777777777707377777777
115    77777777777777777777777777777777777777741737737476373567365633731777777777
116    77777777777777777777777777777777777777721757717436171463175633737777777777
117    77777777777777777777777777777777777777711767747516471463164613713777777777
118    77777777777777777777777777777777777777705773763547634461174753713777777777
119    77777777777777777777777777777777777777037757755627324241667717047777777777
120    61777777777777777777777777777777777777773767765727125370766417277777777777
121    61777777777777777777777777777777777777767773771707421372343717277777777777
122    77777777777777777777777777777777777777777775770747631173250017217777777777
123    77777777777777777777777777777777777777777776770001631573001707677777777777
124    77777777777777777777777777777777777777737741737476373567365633731777777777
125    77777777777777777777777777777777777777757721717436171463175633737777777777
126    77777777777777777777777777777777777777767711747516471463164613713777777777
127    77777777777777777777777777777777777777737057635476344611747537137777777777
128    77777777777777777777777777777777777777757037755627324241667717047777777777
129    77777777777777777777777777777777777777767770765727125370766417267777777777
130    77777777777777777777777777777777777777773764771707421372343717267777777777
131    77777777777777777777777777777777777777775762770747631173250617207777777777
132    77777777777777777777777777777777777777776761770001631573015707667777777777
133    77777777777777777777777777777777777777717717741476373563375633733777777777
134    77777777777777777777777777777777777777747747721436171473175633733777777777
135    77777777777777777777777777777777777777763763711516471463374613717777777777
136    77777777777777777777777777777777777777737737055476344611747537137777777777
137    77777777777777777777777777777777777777757757035627324241767717047777777777
138    77777777777777777777777777773777777777776776777172712537062641727777777777
139    77777777777777777777777777773777777777777077076470742117200160723777777777
140    77777777777777777777777777777773777770000000002017775473014707307777777777
141    77777777777777777777777777777773777777174174171017775433014307317777777777
142    77777777777777777777777777777773777777476476474417774531215107137777777777
143    77777777777777777777777777777773777777636636634217756515025507537777777777
144    77777777777777777777777777777773777777716716714117776145025707467777777777
145    77777777777777777777777777777773777777427427400577771645215476677777777777
146    77777777777777777777777777777773777777707707700377771760316476177777777777
147    77777777777777777777777777777777777777040040043770033673414727307777777777
148    77777777777777777777777777777777777777176176177764033633414337217777777777
149    77777777777777777777777777777777777777476476477762013731515127177777777777
150    77777777777777777777777777777777777777637637637776105771576753753777777777
151    77777777777777777777777777777777777777077077077604477447674574477777777777
152    77777777777777777777777777777777777777637637637602677763636576777777777777
153    77777777777777777777777777777777777777077077376016377703174765777777777777
154    77777777777777774377777777777777763777040040043761743633656337307777777777
155    77777777777777774377777777777777763777437437437574323715747137137777777777
156    77777777777777774377777777777777763777703703703777013744363457447777777777
157    77777777777777774377777777777777763777770770773761607777031747657777777777
158    77777777777777774377777777777777774777000000000037777603165633730777777777
159    77777777777777774377777777777777774777417417417077775115647117127777777777
160    77777777777777774377777777777777774777741741743617774544263457667777777777
161    77777777777777774377777777777777774777774774770017774377031647617777777777
```

OUTPUT

Figure 6 (cont.)

```
162   776377777777777777777777777777777777777701701721476373570775633732777777777
163   77637777777777777777777777777777777777437437036170714647667137137777777777
164   776377777777777777777777777777777777777177177174763462276675174677777777
165   776377777777777777777777777777777777777577577576373672177777576677777777
166   777777777777777777777777377777777777770000000003617357746563373077777777
167   777777777777777777777777377777777777774074075070343443726471170077777777
168   777777777777777777777777377777777777777637637637171673717634576577777777
169   777777777777777777777777777607777777777707707700574631774316476377777777
170   777777777777777777777777777607777777777767767704016735773357676777777777
171   777777777777777777777777777773777777772542547600436301113104172077777777
172   777777777777777777777777777773777777777767767600016735770757476777777777
173   777777777777777777777777777777777777770300300303617146716617373177777777
174   777777777777777777777777777777777777776026036034361463147517174377777777
175   777777777777777777777777777777777777777707717737174631762457576177777777
176   777777777777777777777777777777777777777747747700016735770037476577777777
177   777777777777777777777777777777777777770650650647637356736770373377777777
178   777777777777777777777777777777777777770750770763717146317764371377777777
179   777777777777777777777777777777777777776255276271343463146762174777777777
180   777777777777777777777777777777777777777657657637571671762761576377777777
181   777777777777777777741777777777777777761000000000060714611646140747777777
182   777777777777777777741777777777777777751740740740017062144221420747777777
183   777777777777777777741777777777777777751775776770000567357713571077777777
184   777777777777777777740777777777777777760777777777777777777777777704737777777
185   777777777777777777740777777777777777760777777777777777777777777702737777777
186   777777777777777777740777777777777777760777777777777777777777777701737777777
187   777777777777777777740377777777777770170010010003617146316563774477777777
188   777777777777777777740377777777777770176176176170743467157671772477777777
189   777777777777777777740377777777777770177407407436130631440214771477777777
190   777777777777777777776777777777777777762352351301427144312422707577777777
191   777777777777777777777777377777777777773553557547432215706517170277777777
192   777777777777777777777777777777777777777577407556273242416777170577777777
```

OUTPUT

Figure 6 (cont.)

```
                                        PRIORITY  LIST
  COURSE        CONFLICT         DAYS PER        UNITS PER       UNITS PER          BUFFER
 SECTIONS       COUNTER            CYCLE             DAY      DAY CONSTANT     REQUIREMENT

     181            441                5               1               1               0
     182            441                5               1               1               0
     114            347                5               1               1               0
      63            264                5               1               1               0
       6            263                5               1               1               0
      69            257                5               1               1               0
      64            256                5               1               1               0
      67            254                5               1               1               0
      72            254                5               1               1               0
      12            252                5               1               1               0
      70            239                5               1               1               0
      65            238                5               1               1               0
       5            236                5               1               1               0
      71            236                5               1               1               0
       7            230                5               1               1               0
       1            229                5               1               1               0
      66            229                5               1               1               0
       4            227                5               1               1               0
      11            225                5               1               1               0
      10            221                5               1               1               0
     133            214                5               1               1               0
     171            214                5               1               1               0
     176            214                5               1               1               0
       8            212                5               1               1               0
       2            211                5               1               1               0
       3            209                5               1               1               0
      68            209                5               1               1               0
     147            209                5               1               1               0
     155            209                5               1               1               0
     159            207                5               1               1               0
       9            203                5               1               1               0
      73            202                5               1               1               0
     162            201                5               1               1               0
     163            201                5               1               1               0
     164            201                5               1               1               0
     167            201                5               1               1               0
     115            200                5               1               1               0
     116            200                5               1               1               0
     117            200                5               1               1               0
     124            200                5               1               1               0
     125            200                5               1               1               0
     126            200                5               1               1               0
     134            200                5               1               1               0
     166            200                5               1               1               0
     167            200                5               1               1               0
     168            200                5               1               1               0
     173            200                5               1               1               0
     174            200                5               1               1               0
     175            200                5               1               1               0
     177            200                5               1               1               0
     178            200                5               1               1               0
     179            200                5               1               1               0
     189            195                5               1               1               0
     121            193                5               1               1               0
     130            193                5               1               1               0
```

OUTPUT

Figure 6 (cont.)

```
123   186   5   1   1   0
132   186   5   1   1   0
118   179   5   1   1   0
119   179   5   1   1   0
127   179   5   1   1   0
128   179   5   1   1   0
120   172   5   1   1   0
129   172   5   1   1   0
135   172   5   1   1   0
136   172   5   1   1   0
138   172   5   1   1   0
192   172   5   1   1   0
137   165   5   1   1   0
122   151   5   1   1   0
131   151   5   1   1   0
172   144   5   1   1   0
190   124   5   1   1   0
191   123   5   1   1   0
183   112   5   1   1   0
170   103   5   1   1   0
180    74   5   1   1   0
165    40   5   1   1   0
184    17   5   1   1   0
185    17   5   1   1   0
186    17   5   1   1   0
 30   267   4   1   1   0
 84   266   4   1   1   0
 24   263   4   1   1   0
 18   258   4   1   1   0
 79   255   4   1   1   0
 74   257   4   1   1   0
 77   249   4   1   1   0
 82   249   4   1   1   0
 67   248   4   1   1   0
 75   240   4   1   1   0
 80   240   4   1   1   0
 23   236   4   1   1   0
 19   231   4   1   1   0
 25   231   4   1   1   0
 76   231   4   1   1   0
 81   231   4   1   1   0
 13   230   4   1   1   0
 17   230   4   1   1   0
 85   230   4   1   1   0
 86   230   4   1   1   0
 22   222   4   1   1   0
 28   222   4   1   1   0
 29   222   4   1   1   0
 16   221   4   1   1   0
 20   213   4   1   1   0
 26   213   4   1   1   0
 14   212   4   1   1   0
 21   209   4   1   1   0
 27   204   4   1   1   0
 78   204   4   1   1   0
 83   204   4   1   1   0
 15   203   4   1   1   0
 88   202   4   1   1   0
 59   561   2   1   1   0
111   561   2   1   1   0
```

OUTPUT

Figure 6 (cont.)

```
 60   525   2   1   1   0
112   291   2   1   1   0
 53   265   2   1   1   0
 54   265   2   1   1   0
 55   265   2   1   1   0
 56   265   2   1   1   0
 57   265   2   1   1   0
106   265   2   1   1   0
107   265   2   1   1   0
108   265   2   1   1   0
109   265   2   1   1   0
 36   263   2   1   1   0
 89   263   2   1   1   0
 92   254   2   1   1   0
 90   245   2   1   1   0
 31   236   2   1   1   0
 35   236   2   1   1   0
 91   236   2   1   1   0
 34   227   2   1   1   0
 32   218   2   1   1   0
 33   209   2   1   1   0
 93   209   2   1   1   0
 97   177   2   2   2   0
 96   161   2   2   2   0
110   157   2   1   1   0
 94   152   2   2   2   0
 45   152   2   2   2   0
101   152   2   2   2   0
 98   143   2   2   2   0
 99   143   2   2   2   0
100   143   2   2   2   0
140   139   2   2   2   0
141   139   2   2   2   0
143   139   2   2   2   0
144   139   2   2   2   0
146   139   2   2   2   0
142   132   2   2   2   0
145   132   2   2   2   0
147   118   2   2   2   0
148   118   2   2   2   0
149   118   2   2   2   0
150   111   2   2   2   0
151   111   2   2   2   0
152   111   2   2   2   0
153   111   2   2   2   0
 58    76   2   1   1   0
 61    21   2   1   1   0
 62    21   2   1   1   0
113    21   2   1   1   0
104   314   1   2   2   0
105   305   1   2   2   0
 47   296   1   2   2   0
 48   296   1   2   2   0
 50   296   1   2   2   0
 51   296   1   2   2   0
102   296   1   2   2   0
103   269   1   2   2   0
158   239   1   2   2   0
154   232   1   2   2   0
155   232   1   2   2   0
```

OUTPUT

Figure 6 (cont.)

```
         156            232             1              2              2              0
         159            232             1              2              2              0
         160            232             1              2              2              0
         161            190             1              2              2              0
          37            143             1              2              2              0
          38            143             1              2              2              0
          39            143             1              2              2              0
          40            143             1              2              2              0
          42            143             1              2              2              0
          43            143             1              2              2              0
          44            143             1              2              2              0
          45            143             1              2              2              0
          41            134             1              2              2              0
          46            125             1              2              2              0
         157             99             1              2              2              0
          52             98             1              2              2              0
          49             89             1              2              2              0
28    51     0    33     80
```

OUTPUT

Figure 6 (cont.)

```
                                          MASTER SCHEDULE DAY        1

PERIOD

      1       181   114     6    70     1    11    10   176   187   168    76    26    21
      2       182    63    69    67    65   133   188     9    73   170    81    22    20
      3        64    72    12     7   171   162   134   174   183    75    17    86    28    83    32
      4         5    66     8   139   163   115   167   184    84    18    77    25    16    33   110
      5        71     2   164   169   116   124   120   135   185    30    79    19    15    88    56    92
      6         4   117   125   175   132   136   138   165   186    24    74    82    85    29    54    91
      7         3   126   189   118    87    80    13    78    36   141    61    40
      8        40   141    68   179   121   119    14    27   107    31    62

                                 IT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICTS

SECTION   CONFLICTS WITH COURSE SECTIONS

    166        73   133
    173       133   166
    177       115
    178       181   187
    130       174
    125       134
    127       179
    128       167
    129       120
    192       189
    137       119
    122       171
    131       141
    172       176
    190       114   178
    191       124
    180       182

                                          REVISED MASTER SCHEDULE DAY        1

PERIOD

      1       181   114     6    70     1    11    10   176   187   168    76    26    21   178   172   190
      2       182    63    69    67    65   133   188     9    73   170    81    22    20   166   173   180
      3        64    72    12     7   171   162   134   174   183    75    17    86    28    83    32   130   122
      4         5    66     8   139   163   115   167   184    84    18    77    25    16    33   110   177   123   128

      5        71     2   164   169   116   124   120   135   185    30    79    19    15    88    56    92   129   191

      6         4   117   125   175   132   136   138   165   186    24    74    82    85    29    54    91
      7         3   126   189   118    87    80    13    78    36   141    61    40   192
      8        40   141    68   179   121   119    14    27   107    31    62   127   137   131
```

```
                                      MASTER SCHEDULE DAY        2
PERIOD

   1      181  114    6   70    1   11   10  176  187  168   76   26   21
   2      182   63   69   67   65  133  188    9   73  170   81   22   20
   3       64   72   12    7  171  162  134  174  163   75   17   86   28   83   32
   4        5   66    8  139  163  115  167  184   84   18   77   25   16   33  110
   5       71    2  164  169  116  124  120  135  185   30   79   19   15   88   56   92
   6        4  117  125  175  132  136  138  165  186   24   74   82   85   29   54   91
   7        3  126  189  118   87   60   13   78   36   61
   8       68  166  179  172   23   14   27  107   31   62

                        IT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICTS

SECTION   CONFLICTS WITH COURSE SECTIONS

  173       133
  177       133
  178       181   187
  121       174
  130       139
  123       139
  119       166
  127       181   187
  128       167
  129       174
  192       164
  137       189
  122       171
  131       166
  190       131   166
  191       175
  180       139

                                  REVISED MASTER SCHEDULE DAY        2
PERIOD

   1      181  114    6   70    1   11   10  176  187  168   76   26   21  178  127
   2      182   63   69   67   65  133  188    9   73  170   61   22   20  173  177
   3       64   72   12    7  171  162  134  174  183   75   17   86   28   83   32  121  129  122

   4        5   66    8  139  163  115  167  184   84   18   77   25   16   33  110  130  123  128
          180
   5       71    2  164  169  116  124  120  135  185   30   79   19   15   88   56   92  192
   6        4  117  125  175  132  136  138  165  186   24   74   82   85   29   54   91  191
   7        3  126  189  118   87   60   13   78   36   61  137
   8       68  166  179  172   23   14   27  107   31   62  119  131  190
```

OUTPUT

Figure 6 (cont.)

OUTPUT

Figure 6 (cont.)

```
                                          MASTER SCHEDULE DAY        3

PERIOD

   1      181  114    6   70    1   11   10  176  187  168   76   26   21
   2      182   63   69   67   65  133  188    9   73  170   81   22   20
   3       64   72   12    7  171  162  134  174  183   75   17   86   28   83   38
   4       38    5   66  139  163  115  167  184   84   24   82   25   16   93
   5       71    8  164  169  116  124  120  135  185   30   79   77   13   88   55   90
   6        4    2  117  125  175  132  136  138  165  186   18   74   87   19   29
   7        3  126  189  118   80   14   78   89  100  141  113   41   49
   8       41   49  100  141   68  179  121  119   85   27   34

                          IT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICTS

SECTION   CONFLICTS WITH COURSE SECTIONS

   166       73  133
   173      133  166
   177      115
   178      161  187
   130      174
   123      139
   127      179
   128      167
   129      120
   192      189
   137      119
   122      171
   131      141
   172      176
   190      114  178
   191      124
   180      182
    23       41

                                     REVISED MASTER SCHEDULE DAY        3

PERIOD

   1      181  114    6   70    1   11   10  176  187  168   76   26   21  178  172  190
   2      182   63   69   67   65  133  188    9   73  170   81   22   20  166  173  180
   3       64   72   12    7  171  162  134  174  183   75   17   86   28   83   38  130  122
   4       38    5   66  139  163  115  167  184   84   24   82   25   16   93  177  123  128
   5       71    8  164  169  115  124  120  135  185   30   79   77   13   88   55   90  129  191

   6        4    2  117  125  175  132  136  138  165  186   18 . 74   87   19   29
   7        3  126  189  118   80   14   78   89  100  141  113   41   49  192
   8       41   49  100  141   68  179  121  119   85   27   34  127  137  131   23
```

```
                                        MASTER SCHEDULE DAY       4

  PERIOD

     1      131   114     5    70     1    11    10   176   157   168    76    26    21
     2      142    63    69    67    65   133   164     9    73   170    81    22    20
     3       64    72    12     7   171   162   134   174   163    75    17    60    28    83
     4        5    66     9   139   163   115   167   184    44    18    77    25    16    93
     5       71     2   164   169   116   124   120   135   165    30    79    19    15    88    90
     6        4   117   125   175   132   136   138   165   166    24    74    82    85    29    43
     7       43     3   126   189   113    87    80    78    89   113    52
     8       52    68   156   179   172    13    27   108    34

                              IT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICTS

SECTION   CONFLICTS WITH COURSE SECTIONS

   173       133
   177       133
   178       126
   121       174
   130       139
   123       139
   119       166
   127       181   187
   128       157
   129       174
   192       154
   137       183
   122       171
   131       189
   190       166
   191       175
   160       158
    23        52
    59         3    43   113
   111        68   108
    60        15   185
   112       186
    53         7    17    28
    57        63   188
   106        74    85
   109        88
    35        29
    47       114
    97        57    73
    96        62   112
    96        78    87
    94        66    84    93
    94        71    79    88    90
    95        70    76    97
    95        65    69    73    81    97
   101       114
   101        67    73
    98        66    84    93
    98        71    79    88    90
    99        70    76   101
    99        65    69    81   101
   140       118   126   131   137
   140       119   166   172   190
```

OUTPUT

Figure 6 (cont.)

OUTPUT

Figure 6 (cont.)

```
     143     171   174   183
     143     123   163   167
     144     164   169   192
     144     132   136   165
     146     144   169
     146     132   144   175   191
     142     126   140   178
     142     140   166   172   179   190
     145      97   168   176
     145      97   170   182
     147     176   181   187
     147     133   170   173   177
     143     116   124
     148     125   132   191
     149     134   183
     149     123   163   167
     150     147   188
     150     149   174
     151     128   149   167
     151     120   148   164   192
     152     189
     152
     153     147   176
     153     147   150   170   182
      56       5    18

                                     REVISED MASTER SCHEDULE DAY        4

  PERIOD

       1     161   114     6    70     1    11    10   176   167   168    76    26    21   127   180    97    95   101
              99   145   147   153
       2     162    63    69    67    65   133   188     9    73   170    81    22    20   173   177    57    97    95
             101    99   145   147   150   153
       3      64    72    12     7   171   162   134   174   183    75    17    86    28    83   121   129   122    53
             143   149   150
       4       5    66     8   139   163   115   167   184    84    18    77    25    16    93   130   123   128    94
              78   143   149   151    58
       5      71     2   164   169   116   124   120   135   185    30    79    19    15    88    90   192    60   109
              94    98   144   146   148   151
       6       4   117   125   175   132   136   138   165   186    24    74    82    85    29    43   191   112   106
              35    96   144   146   148     .
       7      43     3   126   189   116    67    60    78    69   113    52   178   137   131    59    96   140   142
             152
       8      52    68   166   179   172    13    27   108    34   119   190    23   111   140   142   152
```

OUTPUT

Figure 6 (cont.)

```
                                        MASTER SCHEDULE DAY        5

PERIOD

     1      181   114     6    70     1    11    10   176   187   168    14
     2      182    63    69    67    65   133   188     9    73   170
     3       64    72    12     7   171   162   134   174   183    15    35
     4        5    66     8   139   163   115   167   184    55    97   101
     5       37   101    71     2   164   169   116   124   120   135   185    23
     6        4   117   125   175   132   136   138   165   156    57    96    98   103
     7       95    38   103     3   126   189   119
     8       66   166   177   172    59   108

                              IT IS NECESSARY TO PERMIT THE FOLLOWING CONFLICTS

SECTION   CONFLICTS WITH COURSE SECTIONS

   173       133
   177       133
   178       126
   121       174
   130       139
   123       139
   119       166
   127       181   187
   128       167
   129       174
   192       164
   137       159
   122       171
   131       189
   190       166
   191       175
   180       168
   111        70   181
    60         3
   112       163
    53         2    23
   106        64
   109       114   187
    94        36
    94        36
    95        70   111
    95        65    69    73
    93        64
    99        56   101
   100        67
   100        72    99   112
   140       127   176   161   187
   140       133   170   173   177   182
   143       118   126
   143       172   179   190
   144       171   174   183
   144        97   123   128   167
   146        97   169
   146       132   175   191
   142       126   143   178
   142       143   166   172   179   190
   145       140   168   176
   145       140   170   182
```

OUTPUT

Figure 6 (cont.)

```
   147     176   181   187
   147     133   170   173   177
   148     116   124
   148     125   132   191
   149     134   183
   149     123   163   167
   150     147   188
   150     149   174
   151     128   149   167
   151     120   148   164   192
   152     189
   152
   153     147   176
   153     147   150   170   182
    56      57
   104      66    97
   104      71    97
   105     109   114
   105      67    73
    47       9    63
    47       7    15    35
    48      23
    48       4    57   103
    50       4
    50       3    60
    51      23   104
    51       4    50    57
   102      66    99
   102      48    71
   158     105   127   140   181   187
   158     105   133   140   173   177
   154     147   181   187
   154      47   133   147   173   177
   155     127   154   181   187
   155      47   150   154   173   188
   155      47   121   129   174
   156     102   128   130   139   151   167
   157     134   144   171   174
   159     104   128   144   163   167
   160     105   145   158   182
   160     121   122   129   144   159   171   174
   161      51   104   146   169          .
   161      50    51   132   146   175   191
    37       9    63    95
    37       7    15    35
    39      10    95
    39       9    37    95
    42       4    98
    42       3    98
    44      10    14
    44       9    47   100
    45      44    63   100
    45      12    35    99   100
    46       6    44
    46      44    45    63   100
   157     103   131   189
   157     172

                         REVISED MASTER SCHEDULE DAY      5
```

OUTPUT

Figure 6 (cont.)

```
PERIOD

   1       181   114     6    70     1    11    10   176   187   168    14   127   180   111   109    95   140   145
           147   153   105   158   154   155    39    44    46
   2       182    63    69    67    65   133   188     9    73   170   173   177    95   100   140   145   147   150
           153   105    47   158   154   155   160    37    39    44    45    46
   3        64    72    12     7   171   162   134   174   183    15    35   121   129   122   112   106    99   100
           144   149   150    47   156   159   160    37    45
   4         5    66     8   139   163   115   167   184    55    97   101   130   123   128    99   144   149   151
           104   102   156   159
   5        97   101    71     2   164   169   116   124   120   135   185    23   192    53   146   148   151   104
            48    51   102   161
   6         4   117   125   175   132   136   138   165   186    57    96    98   103   191    94   146   148    58
            48    50    51   161    42
   7        96    98   103     3   126   169   118   178   137   131    60    94   143   142   152    50    42   157

   8        68   166   179   172    59   108   119   190   143   142   152   157
```

OUTPUT

Figure 6 (cont.)

```
NUMBER OF DAYS=     5     PERIOD=     9
0000000000000000000000000000000000000000000000000000000000000000000000
  1641
```

SUGGESTED EXTENSIONS
IN THE
CONTROLLED SCHEDULE
CYCLE CONCEPT

BY

I.B. TURKSEN

A.G. HOLZMAN

FURTHER RESEARCH

The results of the controlled schedule cycle are 1) a weekly master schedule; and 2) a tabulation of conflicts that are forced into each schedule day.

When a course section A_k is forced into the i-th day of the cycle, the course sections that are in conflict with A_k are indicated in an array. If A_k is in conflict with A_a, A_b, A_c for the i-th day, then the array is:

A_k: A_a A_b A_c

If A_k is also forced into the j-th day of the cycle and if A_k is in conflict with A_d, A_b, A_e for this day, then the array is:

A_k: A_d A_b A_e

The paired conflict (A_k, A_b) appears in both arrays.

If A_k is a course section which requires consecutive periods of a session, and if A_k is forced into a period i and period i + 1, then two arrays will be shown. If A_k is in conflict with A_a, A_b, A_c in period i, then

A_k: A_a A_b A_c

If A_k is in conflict with A_b, A_d, A_c in period i + 1, then

A_k: A_b A_d A_c

In this case the paired conflict (A_k, A_b) appears in different arrays of the same schedule day.

In general a paired conflict (A_k, A_b) will appear in different arrays if the conflict (A_k, A_b) is created in more than one schedule day. An organized matrix of forced conflicts should be generated from the results of the controlled schedule cycle algorithm. This new matrix would be used for further investigation.

This new matrix should be called the Forced Conflicts Matrix. This matrix is very similar to the Relations Matrix. The A-Identifiers are the row and column headings. An element, fc_{ij}, of the Forced Conflicts matrix is 0 if Ai and Aj are conflicting in the master schedule and $fc_{ij} = 1$ if Ai and Aj are not conflicting in the master schedule. The diagonal elements $fc_{ii} = 1$ for all i. Clearly, the Forced Conflicts matrix, FC, is a symmetric square matrix.

The Forced Conflicts matrix could be used to separate the faculty, students and physical plant facilities that have conflicting schedules. This could be accomplished by a search technique. The request blocks that have at least one conflict would be submitted to an 'after-assignment' phase.

The 'after-assignment' phase should consist of two steps.

1. Assignment of a new section
2. Assignment of a new course.

If A_k is a conflicting section in a request block, then a) find the μ-identifier of A_k, let it be μ_i; and b) attempt to assign a different section of μ_i which will eliminate a conflict caused by A_k. If no other section of μ_i could be assigned, then investigate the Priority List of course requests for this request block.* Select the next high priority course, let it be μ_j. Attempt to assign a section of μ_j by investigating the Forced Conflicts Matrix.

The after-assignment is practical only if the number of request blocks with at least one conflict is fewer than the total number of request blocks. This will generally be the case.

IMPROVEMENT OF THE CRITERIA WHICH ARE USED TO MINIMIZE THE NUMBER OF CONFLICTS

The idea conceived by P.J. Kiviat could be applied for the selection of the periods when a course section A_k is to be forced into the schedule day. The application is possible with new concepts and interpretation.

Generate a Conflict Matrix, CM, in Phase III of the report. An element CM(i,j) of the conflict matrix indicates the number of times the paired conflict (A_i, A_j) occurs. If r(i,j) of the Relations Matrix is zero, and r(i,j) = 0 was repeated 10 times then CM(i,j) = 10, and CM(j,i) = 10. The diagonal of CM = (0,0,...,0). That is an element CM(i,i) = 0 since no paired conflict of the form $(A_i \cdot A_i)$ can exist. The Conflict Matrix is a symmetric square matrix of order nxn. CM is similar to the P matrix of Kiviat, but it is generated distinctly. The conditional probability matrix, P', is developed by formulations as follows:

---

*Refer to 'Routine to Generate Course Requests and Provide File Maintenance for Student Records' by W. Baughman.

$P'_{ii} = 0$ since no A_i can have a conditional probability of conflicting with itself.

$$P'_{ij} = \frac{P_{ij}}{\sum_{j:j \neq i} P_{ij}}$$

In forcing A_k into a particular day, for each period of this day, determine the course sections that will conflict with A_k. Determine the sum of all the conditional probabilities over this set of course sections. Then order the periods with respect to these probabilistic values. Next select the suitable u_k periods and force A_k into those periods.

For example, consider a forced scheduling situation where:

1. A_k is to be forced into the i-th day;
2. If A_k is forced into the first period, it will conflict with A_a, A_b, A_c;
3. If A_k is forced into the second period, it will conflict with A_d, A_b;

.

.

.

m + 1. if A_k is forced into the m-th period, A_k will conflict A_p, A_r, A_s, A_t

Determine the union of the conditional probabilities for each period.

period 1

$$\begin{aligned} S_1 &= P'_{ka} \cup P'_{kb} \cup P'_{kc} \\ &= P'_{ka} + P'_{kb} + P'_{kc} - P'_{ka} \cdot P'_{kb} - P'_{ka} \cdot P'_{kc} \\ &\quad - P'_{kb} \cdot P'_{kc} + P'_{ka} \cdot P'_{kb} \cdot P'_{kc} \end{aligned}$$

period 2

$$\begin{aligned} S_2 &= P'_{kd} \cup P'_{kb} \\ &= P'_{kd} + P'_{kb} - P'_{kd} \cdot P'_{kb} \end{aligned}$$

•

•

•

period m

$$\begin{aligned} S_m &= P'_{kp} \cup P'_{kr} \cup P'_{ks} \cup P'_{kt} \\ &= P'_{kp} + P'_{kr} + P'_{ks} + P'_{kt} \\ &\quad - P'_{kp} \cdot P'_{kr} - P'_{kp} \cdot P'_{ks} - P'_{kp} \cdot P'_{kt} \\ &\quad - P'_{kr} \cdot P'_{ks} - P'_{kr} \cdot P'_{kt} - P'_{ks} \cdot P_{kt} \\ &\quad + P'_{kp} \cdot P'_{kr} \cdot P'_{ks} + P'_{kp} \cdot P'_{kr} \cdot P'_{kt} \\ &\quad + P'_{kp} \cdot P'_{ks} \cdot P'_{kt} + P'_{kr} \cdot P'_{ks} \cdot P'_{kt} \\ &\quad - P'_{kp} \cdot P'_{kr} \cdot P'_{ks} \cdot P'_{kt} \end{aligned}$$

The periods are ordered with respect to the values of the elements of $S = (S_1, S_2, \ldots, S_m)$. Period i is ordered before period i' if $S_i < S_{i'}$. This new period priority list may now be used for the selection of the periods. This criterion would guarantee the most likely minimum number of conflicts. The dimensions implied by such an application may jeopardize the practicality.* However, theoretically, this probabilistic approach does have merit, and should be investigated further.

FURTHER IMPLICATIONS OF CONDITIONAL PROBABILITY CONCEPT

The matrices CM and P' could in a dynamic sense be used in Phase II of the report. In Phase II, after each assignment an element of CM & P' will be changed. Each assignment could be made on the basis of the previous conditional probabilities. In this case, however, maximum P'_{ij} is selected for a fixed i among j.

Again, problems of dimensionality exist.

---

*A 1000 course section problem require a CM matrix of 1000 x 1000 = 10^6 computer words.

PROCEDURE, ANALYSES OF DATA, AND FINDINGS

LINEAR PROGRAMMING SCHEDULER

THE LINEAR PROGRAMMING APPROACH TO MASTER TIME SCHEDULE GENERATION IN EDUCATION

BY

ROBERT E. HARDING

INTRODUCTION

The following work outlines a basically different approach to the problem. It builds a general linear mathematical model of the master time schedule restraining relations and makes some initial steps to operationally solving the large system so generated. The approach is to use the methods known as linear programming and examples are presented for clearer understanding. There is much work yet unfinished in developing the potential of this particular approach to scheduling and a detailed statement of some additional research needs along this line are presented.

THE GENERAL LINEAR MODEL

We have J blocks of time equal in length, chronologically numbered from 1 to J with block 1 as the first in the schedule cycle* and block J as the last. It is not necessary that all periods be immediately adjacent in real time. For example, if the schedule cycle is one five day week and the first time block available for scheduling is 8-9 AM Monday, this period would be period number one. If there are nine one-hour blocks on Monday and the last one is 4-5 PM then this would be block 9 and the 8-9 AM period on Tuesday would be number 10. Finally, block 45 would be 4-5 PM Friday. We say, in this case, that the schedule cycle is 45 periods long. We also recognize that there may be sub-cycles within the schedule cycle. Such sub-cycles will most often be days of the week although other possibilities do present themselves. In the example above, we may describe the sub-cycle as being nine periods in length. In general, each schedule cycle may have n sub-cycles where n is arbitrarily an integer which may range from 1 up. Hence, the length of a sub-cycle is J/n periods.

A given educational institution will have a specific number of course section offerings which must be scheduled. We will denote this number by I and arbitrarily assign number-names to all I starting with 1 and going to I.

---

*See Glossary for specialized usage of terms.

Note that a schedule for a course section is a series of assigned time periods. We might think of a number of slots representing J time periods. We have a set of such slots for each course section and assign either 0 or 1 to each slot: 1 denoting that the course section is scheduled for that time period, 0 that it is not so scheduled. Let X_{ij} represent the value assigned to the jth slot of the ith course section. If i goes from 1 to I and j goes from 1 to J, we have I x J slots to which we must assign values.

PERIODS OF THE CYCLE

$$\begin{array}{llcccccc}
 & & 1 & 2 & \cdots & j & \cdots & J \\
\text{Course} & 1 & X_{11} & X_{12} & \cdots & X_{1j} & \cdots & X_{1j} \\
 & 2 & X_{21} & X_{22} & \cdots & X_{2j} & \cdots & X_{2J} \\
\text{Section} & \vdots & \vdots & \vdots & & \vdots & & \vdots \\
 & i & X_{i1} & X_{i2} & \cdots & X_{ij} & \cdots & X_{iJ} \\
\text{(Outer)} & \vdots & & & & & & \\
 & I & X_{I1} & X_{I2} & \cdots & X_{Ij} & \cdots & X_{IJ}
\end{array}$$

Figure 1

Our model has I x J real variables.

There are particular relationships which must exist among the variables in our model. These we will describe as side relations or constraints and are the particular expressions which in fact 'model' the real phenomenon in question.

There will be listed below all of the common types of such relations which are found in educational institutions. Both the verbal and the symbolic statement of each type of side relation will be presented.

THE MODEL

Variables: X_{ij} $i = 1, \ldots, I$

$j = 1, \ldots, J$

With the following relations holding

1. $X_{ij} \leq 1$ for all i and j. There are I x J of these inequalities which tell us that we can deliver a given time period to a specific course section no more than once. A course section i can meet no more than once in a given time period j.

2. $\sum_{j=1}^{J} X_{ij} = T_i$ There are I of these equality relations each indicating that a specific course section, the ith, must meet exactly T_i time units per schedule cycle. The numbers T_i will themselves be functions of the schedule life,* the schedule cycle, and the distribution of credits expected to be offered per schedule life. For example, in a university where the schedule cycle is one week and the schedule life is 15 or 16 weeks, typical T_i's would be 1, 2, 3, 4, or 5 time units per schedule cycle, with 3 and 4 being the most frequently observed. We note that $T_i \times \frac{\text{schedule life}}{\text{schedule cycle}} =$ total

---

*See Glossary

scheduled time periods per schedule life for course section i.

3a. $X_{ij} = 0$ These are constraints which indicate that a particular time period (the jth) is not to be used in a schedule for a particular course section (the ith). The number of these constraints will vary from institution to institution. In finding a solution to our problem -- generating a master-time schedule -- there are alternative ways for handling constraints of types 3a and 4d. These will be described below in the discussion of approaches to solving the system.

3b. If the administration would like to prohibit the use of a particular time unit in the schedule of any course section in a set K it can write a single constraint to do this. $\sum_{i \epsilon K} X_{ij} = 0$ which says do not use the jth time unit in schedules for any of the courses in the set K. The use of such restraints is, of course, at the discretion of each school administration.

3c. In this type of restraint, the administration could require that a particular period in each sub-cycle not be used for any of the course sections in a set K. A double sum of the following form would symbolically state this.

$\sum_{i \epsilon K} \sum_{j = p} X_{ij} = 0$ where the index j runs from $p \leqslant J/n$ (the pth period in the first sub-cycle) in steps of J/n on up through the last sub-cycle. Thus $j = p, J/n + p, 2\,J/n + p, \ldots, (n - 1)\,J/n + p$.

This type of constraint insures that the pth period of each sub-cycle is not used for any of the course sections in the set K.

4a. $X_{ij} = 1$ Constraints of this type indicate that the ith course section must use period j in its schedule. The number of these will also vary from institution to institution.

4b. If the administration for some reason desires all course sections in a set K to have some particular time period scheduled in common, the jth, the following constraint, together with those described above in (1) will insure this.

$\sum_{i \epsilon K} X_{ij} = k$ where k is equal to the number of course sections in K -- the total number of course sections in set K which are scheduled for time period j must equal k, the number of course sections in the set. A given time block can be included in a given course section no more than once because of constraint one. This insures that each course section in the set K is assigned the jth time period exactly once. It is entirely possible, of course, that one set of course sections all requiring a specific time period in common, have a subset which need some additional period together. A simple application of the above constraint to the two sets of course sections is indicated in such a situation.

5. Perhaps the most common constraint in educational scheduling concerns the problem of 'conflicts' between course sections. There are three basic reasons for avoiding conflicts between course sections in a set K. (a) Because all k of these courses are taught by the same instructor who can teach only one course section at a time; (b) Because all k of these course sections must be attended by a particular set of students and they can attend only one class section at a time; or (c) Because all k of these course sections must meet in a particular room (e.g. the chemistry lab) and it can accommodate no more than one class at a time.

To insure that no two course sections in the set K are scheduled at the same time we must write J inequality constraints (one for each time period). These will be of the following form:

$\sum_{i \epsilon K} X_{ij} \leq 1$ for j = 1, ..., J each of which tells us that the jth time period can be used no more than once in the schedules for the set of K course sections.

There will be J of these constraints for each set of K course sections which must be scheduled without conflicts.

The reader will note that the analysis of the school data to determine the appropriate constraints of type 5 is a major activity involved in all approaches to scheduling educational institutions.

6. In the same class as constraints of type five are those which indicate that the course sections in the set K not conflict in a particular subset of the J periods of the schedule-cycle. This would be written as above:

$\sum_{i \epsilon k} X_{ij} \leqslant 1$ one constraint for each j in the subset.

7. Another variation of these 'capacity' constraints will be policies of the administration which state that no more than b_t course sections in a set K can overlap (conflict) in any time period. For example, K might be a set of all course sections of English I and there are b_t teachers available for teaching these course sections -- $b_t < k$.

$$\sum_{i \epsilon K} X_{i1} \leq b_t$$
$$\cdots$$
$$\sum_{i \epsilon K} X_{iJ} \leqslant b_t$$

There will be J of these inequalities for each set of course sections such as K. Note that we might also see that b_t is a function of the number of independent schedule groups in the student body taking the courses in the set K.

8. Many institutions set an upper limit on the number of time units from any schedule sub-cycle which may be used for a given course section. For example, if our sub-cycles are days, an administration might require that no course section in a set K use more than L_i time periods from any day. We can write this

symbolically for those i in K which have $T_i > L_i$ (since $T_i \leq L_i$ automatically insures what we are here concerned with) by writing an inequality for each sub-cycle for each $i \epsilon K$ with $T_i > L_i$ as follows;

$$\sum_{j=1}^{J/n} X_{ij} \leq L_i$$

$$\sum_{j=1+J/n}^{2J/n} X_{ij} \leq L_i$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\sum_{j=\frac{n-1}{n}J+1}^{J} X_{ij} \leq L_i$$

Note that for all i in K,L_i is fixed. Thus we have for each set K of such course sections, with k elements, n constraints per course section or n x k constraints per set of course sections. Each set of n guarantees that the ith course section will have no more than L_i time periods in its schedule from any one of the n schedule sub-cycles. It would be possible if all $T_i > L_i$, to have Ixn of these inequalities.

9. An additional kind of general constraint exists in order that administrators might direct that a particular course section use time units from specified sub-cycles without actually indicating the specific schedule. Perhaps the gym teacher is only available on Mondays, Wednesdays, and Fridays. Course section i might be

required to use time blocks from Monday, Wednesday or Friday. Note that there is a subtle difference between this and the requirement that course section i use time periods from Monday, Wednesday, and Friday.

Constraints of type 9 are written as appropriate combinations of inequalities of the following form;

$$\sum_{j = 1 + rJ/n}^{(r + 1)J/n} X_{ij} \geqslant 1 \qquad r \text{ can take on values } 0, 1, \ldots, n - 1$$

This says that course section i be scheduled to use at least 1 time period from the (r + 1)st sub-cycle of the schedule cycle. We could have as many as T_i of these constraints for any course section. If the number is exactly T_i we are naming the sub-cycles from which every time unit must come. If it is less than T_i we are obviously letting at least one of the time units sub-cycle be determined in the solution.

The reader will note that these kinds of constraints do not fix the particular period that will be assigned but only the subsets from which these periods will come. For example, if $T_i = 3$, two constraints could be written so that at least two of course section i's periods are scheduled on specified days -- perhaps Monday and Friday. The third period could fall on any day of the week which would satisfy all the other conditions on the assignment.

In summary, any number (up to I) of course sections could have from 0 to T_i constraints of this type. The maximum number possible is $\sum_{i=1}^{I} T_i$.

10. It is common to find situations which require that, if more than one time period from a sub-cycle is part of the schedule for a given course section, the time units so assigned must be consecutive periods in real time. In other words, administrators may not allow a given course section, say Senior Physics, to meet in two-non-adjacent periods of time of any day of the week. Such requirements require the writing of large numbers of inequality relations. For each of the I course sections the following large set of linear inequalities expresses the above described policies.

$$\left[\begin{array}{llll}
X_{i1} & -X_{i2} & +X_{i3} & \leq 1 \\
X_{i2} & -X_{i3} & +X_{i4} & \leq 1 \\
\cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \\
X_{i((\frac{J}{n})-2)} & -X_{i((\frac{J}{n})-1)} & +X_{i(\frac{J}{n})} & \leq 1
\end{array}\right]$$

$$\left[\begin{array}{llll}
X_{i((J/n)+1)} & -X_{i((\frac{J}{n})+2)} & +X_{i((\frac{J}{n})+3)} & \leq 1 \\
X_{i((\frac{J}{n})+2)} & -X_{i((\frac{J}{n})+3)} & +X_{i((\frac{J}{n})+4)} & \leq 1 \\
\cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \\
X_{i((2J/n)-2)} & -X_{i((2J/n)-1)} & +X_{i((2J/n))} & \leq 1
\end{array}\right]$$

$$\begin{array}{llll}
\cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \\
\cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot
\end{array}$$

$$\left[\begin{array}{llll}
X_{i(((n-1)J/n)+1)} & -X_{i(((n-1)J/n)+2)} & +X_{i(((n-1)J/n)+3)} & \leq 1 \\
\cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot & \cdot \\
X_{i(J-2)} & -X_{i(J-1)} & +X_{iJ} & \leq 1
\end{array}\right]$$

Each of these blocks symbolically states that part of the policy which might verbally be stated as follows: course section i cannot be scheduled in such a way that there are scheduled two periods from a given sub-cycle separated by a period used for some other course section. Each block symbolically states this policy for each sub-cycle in the schedule cycle. Thus, if we have a five day weekly schedule, there would be five blocks as shown above for each course section to which the policy applies. For each course section for each sub-cycle there are in general $(J/n)-2$ of these constraints. Hence, if there are n sub-cycles and if this policy applies to all the course sections then this part of the precisely stated constraint requires, in symbolic form, exactly $I \cdot n\ ((J/n)-2)$ linear inequalities. Removing parentheses this reduces to

$I \cdot n \cdot J/n - 2I \cdot n = I \cdot J - 2In = I(J-2n)$. Of course when $n \geq 1/2J$ we need no such constraints and therefore we must be careful to specify this condition. If N_1 represents the number of such inequalities we can write the following relation:

$N_1 = 0$ when $n \geq 1/2J$

$N_1 = I(J-2n)$ $n < 1/2J$

The reader will observe that these N_1 constraints do not represert the complete policy. Additional large numbers of mathematical inequalities must be written to make the symbolic statement of the policies accurately reflect the actual aims of the administration. Thus we must have, besides those given above, n more groups like the following for each of the I course sections. These say that no assignment of times from any sub-cycle to a given course section will be separated by two time units used for other purposes.

$$X_{i((rJ/n)+1)} \quad -X_{i((rJ/n)+2)} \quad -X_{i((rJ/n)+3)} \quad +X_{i((rJ/n)+4)} \leq 1$$

$$X_{i}{}_{((rJ/n)+2)} \quad -X_{i((rJ/n)+3)} \quad -X_{i((rJ/n)+4)} \quad +X_{i((rJ/n)+5)} \leq 1$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$X_{i(((r+1)J/n)-3)} \quad X_{i(((r+1)J/n)-2)} \quad -X_{i(((r+1)J/n)-1} \quad +X_{i(((r+1)J/n)} \leq 1$$

For a given course section, the ith, we can write this part of the policy mathematically for each schedule sub-cycle by substituting the appropriate value for r. Thus for the kth sub-cycle we insert (k-1) in place of r and the block of restrictions insures that for the kth sub-cycle (and the ith course section) no schedules will be allowed with periods assigned such that there is a two period gap between them.

As we let r go from 0 to n-1 and i from 1 to I in the appropriate fashion we have all the mathematical inequalities for this part of the policy. The number of these inequalities is

$N_2 = I\ (J-3n)$ for $n < 1/3\ J$

$N_2 = 0$ $n \geq 1/3\ J$

We have additional blocks of constraints generated in similar fashion to precisely state the policy against the use of time periods for a given course section and from any schedule sub-cycle in such a way that there are 3, 4, ..., or (J/n)-2 period gaps between them. In the symbolic statement for this last case we write the constraints generally as follows:

$$X_{i1} - X_{i2} - \dots - X_{i((J/n)-1)} + X_{i(J/n)} \leqslant 1$$

$$X_{i((J/n)+1)} - X_{i((J/n)+2)} - \dots - X_{i((2J/n)-1)} + X_{i((2J/n))} \leqslant 1$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$X_{i(((n-1)J/n)+1)} - X_{i(((n-1)J/n)+2)} - \dots - X_{i(((J-1)))} + X_{i(((J)))} \leqslant 1$$

To precisely state this policy with symbols we have to write a total number of linear inequalities which is the sum of N_p, $p = 1, \dots, (J/n)-2$ where

$N_p = I(J-(p+1)n)$ if $n > J/(p+1)$

$N_p = 0$ otherwise, $n \leqslant J/(p+1)$

thus

$$\sum_{p=1}^{p^*} N_p$$ Where p^* is equal to $(J/n)-2$

$$= \sum_{p=1}^{p^*} I(J-(p+1)n)$$

$$= I\sum_{p=1}^{p^*} (J-(p+1)n)$$

$$= I\left(\sum_{p=1}^{p^*} J - \sum_{p=1}^{p^*} n(p+1)\right)$$

$$= IJp^* - n\sum_{p=1}^{p^*} p - n\sum_{p=1}^{p^*} 1$$

$$= IJp^* - \frac{np^*(p^*+1)}{2} - np^*$$

$$= p^*\left(IJ - \frac{n(p^*+1)}{2} - n\right)$$

substituting back in $(J/n) - 2 = p^*$

$$= \quad ((J/n)-2) \quad (IJ-\frac{n((J/n)-2+1)}{2} - n)$$

$$= \quad ((J/n)-2) \quad (IJ-\frac{n((J/n)-1)}{2} - n) = ((J/n)-2) \; (IJ- \frac{J-n}{2} - n) \; .$$

represents the total number of constraints necessary to symbolically express the policy that no course section can be scheduled in such a way that it meets more than once in any schedule sub-cycle with the meetings separated by *one or more* periods of anything else.

11. Another type of constraint represents basic educational philosophy and has been the subject of considerable discussion in the field. The policy is concerned with the basic nature of the course section schedule patterns should they be fixed-period* or mixed schedule patterns.

A fixed-period course section schedule pattern has time periods assigned in such a way that they use blocks from the same position in each schedule sub-cycle. For example, a schedule of M-W-F at 9:00 AM would be a fixed-period schedule pattern. The specific inequalities which will state this policy precisely will be sums of n or J/n terms, (which ever is smaller) one term from each sub-cycle of the schedule cycle and each from a different 'level' in the sub-cycle. The sum of these must be less than or equal to one if the course section is to be scheduled in a fixed period pattern, e.g. if it is more than one the given course section will meet in at least two differently located periods of the sub-cycle (the day) and hence the pattern will not be a fixed period pattern. For a given course section, the ith, the set of such inequalities can be generated as follows: There will be a large set of inequality constraints each with q^* terms where q^* is the smaller of n and J/n. Thus each inequality will have one term from each sub-cycle (if $q^* = n$) or from each position in the sub-cycle (if $q^* = J/n$). Of course, when $q^* = n = J/n$ both of these conditions will hold.

---

*See Glossary

Essentially we say symbolically that the number of time units assigned to a given course section from different levels of different sub-cycles must be less than or equal to one. Each inequality does this for a particular combination of time periods all from different levels of different sub-cycles.

Let us write out a sample set of such constraints for a 'toy' problem. This example has n = 3, J = 12, hence J/n = 4 n = 3. There are more levels of a sub-cycle than there are sub-cycles. In this example the symbolic constraints precisely state the requirement of fixed period type schedules would be as follows for some typical course section, the ith. First remember our numbering system:

Day	1				2				3			
Day Period	1	2	3	4	1	2	3	4	1	2	3	4
Cycle Period	1	2	3	4	5	6	7	8	9	10	11	12

Then constraints for the ith course section schedule to be fixed period type

$$X_{i1} + X_{i6} + X_{i11} \leq 1 \qquad X_{i2} + X_{i5} + X_{i11} \leq 1$$

$$X_{i1} + X_{i6} + X_{i12} \leq 1 \qquad X_{i2} + X_{i5} + X_{i12} \leq 1$$

$$X_{i1} + X_{i7} + X_{i10} \leq 1 \qquad X_{i2} + X_{i7} + X_{i9} \leq 1$$

$$X_{i1} + X_{i7} + X_{i12} \leq 1 \qquad X_{i2} + X_{i7} + X_{i12} \leq 1$$

$$X_{i1} + X_{i8} + X_{i10} \leq 1 \qquad X_{i2} + X_{i8} + X_{i9} \leq 1$$

$$X_{i1} + X_{i8} + X_{i11} \leq 1 \qquad X_{i2} + X_{i8} + X_{i11} \leq 1$$

$$X_{i3} + X_{i5} + X_{i10} \leq 1 \qquad X_{i4} + X_{i5} + X_{i10} \leq 1$$

$$X_{i3} + X_{i5} + X_{i12} \leq 1 \qquad X_{i4} + X_{i5} + X_{i11} \leq 1$$

$$X_{i3} + X_{i6} + X_{i9} \leq 1 \qquad X_{i4} + X_{i6} + X_{i9} \leq 1$$

$$X_{i3} + X_{i6} + X_{i12} \leq 1 \qquad X_{i4} + X_{i6} + X_{i11} \leq 1$$

$$X_{i3} + X_{i8} + X_{i9} \leq 1 \qquad X_{i4} + X_{i7} + X_{i9} \leq 1$$

$$X_{i3} + X_{i8} + X_{i10} \leq 1 \qquad X_{i4} + X_{i7} + X_{i10} \leq 1$$

The number of such constraints is astronomical and is expected to be a major obstacle in making the general model operational for all cases. The number per course section, N*, is ;

$$\frac{(J/n)!}{((J/n)-n)!} \qquad \text{if} \qquad J/n \geq n$$

and

$$\frac{n!}{(n-(J/n))!} \qquad \text{if} \qquad n > J/_n$$

note these are equivalent when n = J/n since $\frac{(J/n)!}{0!} = \frac{n!}{0!}$ when n = J/n.

Because there are I course sections to be scheduled it is possible, if the fixed period policy applies to all I, to have as many of I • N* of these constraints. Obviously if the policy applies to some subset of I the number of constraints will be less than I • N*.

This completes the description of the general model as such and leaves only a discussion of methods for solving for the appropriate values of the variables subject to the set of constraining relations. Depending on the circumstances in the particular situation we may desire values for the X_{ij} which will maximize some criterion function or simply values which do not violate any of the specified constraints. We can formally state the criterion function in general as

$$\sum_{i=1}^{I} \sum_{j=1}^{J} C_{ij} X_{ij}$$

where each C_{ij} is a 'figure of merit' for scheduling course section i to time period j. This may be some measure of desirability or of 'cost' indicating goals of maximizing or minimizing the criterion function, respectively. In the case where criteria are unimportant as long as the schedule satisfies the constraints the C_{ij}'s all have the same value. For simplicity, when this is the case, we will use $C_{ij} = 1$. It is quite obvious that if a solution to a model of this general form exists, it can be found using the computational device for solving linearly constrained problems with linear criterion functions -- linear programming. There is no question that these are solvable problems. The basic task of future research is to resolve the questions of how to efficiently handle the mountain of computations involved with existing computer hardware.

In our research we have demonstrated that this approach is practical for particular cases of the general model by actually generating a master time schedule and its alternatives for a local senior high school.

Below I present two sample applications of this approach, one to a 'toy' problem to familiarize the reader with the basic mode of application and secondly to a real high school -- The Turtle Creek Senior High School of Turtle Creek, Pennsylvania. I would like to personally express here my gratitude to several people whose cooperation made this example possible. Dr. James S. Snoke of the Allegheny County School Department made the initial contact, Dr. Edward C. Cibik, Jr. Supervising Principal of Turtle Creek Schools, the staff of Turtle Creek Senior High School, especially Ralph Mastandrea, counselor and scheduler, who provided much of the data and worked closely with us in making this first actual application, finally to my graduate students whose diligent labors were the prime element in attaining operationality -- Paul M. Dargay, F. Theodore Helmer, Allen W. Kurki, Samuel M. Kublanow, and Charles E. Powell.

Example 1 - A 'Toy' Problem

Because of the small size of this problem all types of constraints will not be exhibited, however a sufficient number will be used to show the reader the general nature of the approach. Suppose in our 'toy' institution we hold classes only three days a week (Monday, Wednesday, and Friday) and have available for scheduling two three hour periods per day 9-12 AM and 1-4 PM. Thus $J = 6$, $n = 3$, and $J/n = 2$. Let us further suppose that a survey of the faculty and student body of our school indicates that an accurate set of measures of the 'desirability' of using each of the six time periods for any course section is as follows:

$$C_{i1} = 2,\ C_{i2} = 3,\ C_{i3} = 2,\ C_{i4} = 2,\ C_{i5} = 2,\ C_{i6} = 1$$
$$\text{for } i = 1, 2, \ldots, I$$

Assume now, that the administration of our school wants to schedule two course sections in such a way that 'desirability' is maximized. The formal statement of this problem can now be given (I will present the verbal statements of the constraints along with their symbolic expressions.

We want to maximize the desirability index

$$D = 2X_{11} + 3X_{12} + 2X_{13} + 2X_{14} + 2X_{15} + 1X_{16}$$
$$+ 2X_{21} + 3X_{22} + 2X_{23} + 2X_{24} + 2X_{25} + 1X_{26}$$

subject to the following constraining relations:

1. $\sum_{j=1}^{6} X_{1j} = 2$ - course section 1 must have exactly 2 time periods

2. $\sum_{j=1}^{6} X_{2j} = 2$ - course section 2 must have exactly 2 time periods

3. $\sum_{i=1}^{2} X_{i4} \leq 1$ - at least one instructor must be free during period 4 (Wednesday afternoon)

4a. $\sum_{j=1}^{2} X_{1j} \leq 1$ - course section 1 cannot meet both morning and afternoon of the same day - on any day.

$\sum_{j=3}^{4} X_{1j} \leq 1$

$\sum_{j=5}^{6} X_{ij} \leq 1$

4b. $\sum_{j=1}^{2} X_{2j} \leq 1$ - two meetings in the same sub-cycle are also prohibited for course section two

$\sum_{j=3}^{4} X_{2j} \leq 1$

$\sum_{j=5}^{6} X_{2j} \leq 1$

5. $X_{11} \leq 1$ $X_{21} \leq 1$
$X_{12} \leq 1$ $X_{22} \leq 1$
$X_{13} \leq 1$ $X_{23} \leq 1$
$X_{14} \leq 1$ $X_{24} \leq 1$
$X_{15} \leq 1$ $X_{25} \leq 1$
$X_{16} \leq 1$ $X_{26} \leq 1$

These insure that no more than 1 unit of a particular time period is delivered to a specific course section.

6. Finally the non-negativity constraints $X_{ij} \geq 0$ $i = 1, 2,$ and $j = 1, 2, \ldots, 6.$

In order to use the Simplex method of linear programming to compute the solution we must first add the appropriate slack variables to make equality constraints of the inequalities and, if an identity matrix to start with is still not available, add the appropriate artificial variables. These will be added with such a large negative value of desirability that we are assured they will be driven out of any solution.

Thus, since we have 19 inequality constraints in this problem we add 19 slack variables $s_1, \ldots, s_{19}$ to make these equalities. The C's associated with these slack variables are all 0. We have two equality constraints, hence must add two artificial variables A_1 and A_2 with which we associate c's indicating it is highly undesirable to use these in any solution. The c's for both, we arbitrarily make -100. To restate the problem as it will be solved by the Simplex method:

Objective: Maximize

$$\begin{aligned} &2X_{11} + 3X_{12} + 2X_{13} + 2X_{14} + 2X_{15} + 1X_{16} \\ &+ 2X_{21} + 3X_{22} + 2X_{23} + 2X_{24} + 2X_{25} + 1X_{26} \\ &- 100A_1 - 100A_2 \end{aligned}$$

subject to:

1. $X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} + A_1 = 2$
2. $X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} + A_2 = 2$
3. $X_{14} + X_{24} + S_1 = 1$
4. $X_{11} + X_{12} + S_2 = 1$
5. $X_{13} + X_{14} + S_3 = 1$
6. $X_{15} + X_{16} + S_4 = 1$
7. $X_{21} + X_{22} + S_5 = 1$
8. $X_{23} + X_{24} + S_6 = 1$

9. $X_{25} + X_{26} + S_7 = 1$
10. $X_{11} + S_8 = 1$
11. $X_{12} + S_9 = 1$
12. $X_{13} + S_{10} = 1$
13. $X_{14} + S_{11} = 1$
14. $X_{15} + S_{12} = 1$
15. $X_{16} + S_{13} = 1$
16. $X_{21} + S_{14} = 1$
17. $X_{22} + S_{15} = 1$
18. $X_{23} + S_{16} = 1$
19. $X_{24} + S_{17} = 1$
20. $X_{25} + S_{18} = 1$
21. $X_{26} + S_{19} = 1$

Thus we have 12 real, 2 artificial, and 19 slack variables; a total of 33 variables. There are 21 constraints.

The Simplex method of linear programming insures that we will find the program (the assignment of values to the X's and S's) which will maximize the criterion function and at the same time satisfy all of the constraints. In addition, it quickly detects any inconsistent constraints and informs the operator of their existence. Finally, and perhaps very valuable for educational scheduling, it informs us of alternative solutions which are as good as the original and provides us with a ready means of finding these other assignments. The computer program solution to this simple problem follows.

INITIAL TABLEAU

			P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33
			2	3	2	2	2	1	2	3	2	2	2	1													-100	-100							
C	B	P0	X11	X12	X13	X14	X15	X16	X21	X22	X23	X24	X25	X26	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	A1	A2	S13	S14	S15	S16	S17	S18	S19
	P13	1	1												1																				
	P14	1		1												1																			
	P15	1			1												1																		
	P16	1				1												1																	
	P17	1					1												1																
	P18	1						1												1															
	P19	1							1												1														
	P20	1								1												1													
	P21	1									1												1												
	P22	1										1												1											
	P23	1											1												1										
	P24	1												1												1									
-100	P25	2	1	1	1	1	1	1																			1								
-100	P26	2							1	1	1	1	1	1														1							
	P27	1				1						1																	1						
	P28	1	1	1																										1					
	P29	1			1	1																									1				
	P30	1					1	1																								1			
	P31	1							1	1																							1		
	P32	1									1	1																						1	
	P33	1											1	1																					1
Zj	-Cj	-400	-102	-103	-102	-102	-102	-101	-102	-103	-102	-102	-102	-101	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FINAL TABLEAU

			2	3	2	2	2	1	2	3	2	2	2	1													-100	-100							
	P13	1	1												1																				
3	P2	1		1												1																			
	P15	0	-1		0	-1	-1	-1			1						1					-1													
	P16	1				1												1																	
	P17	1					1												1																
	P18	1						1												1															
	P19	0							1												1														
3	P8	1								1												1													
	P21	1									1												1												
	P22	0							-1		-1	0	-1	-1								1		1				-1							
	P23	1											1												1										
	P24	1												1												1									
2	P3	1	1	0	1	1	1	1								-1											1								
2	P10	1							1	0	1	1	1	1								-1						1							
	P27	0				1			-1		-1	0	-1	-1								1						-1	1						
	P28	0	1	0												-1														1					
	P29	0	-1		0	0	-1	-1			1											-1									1				
	P30	1					1	1																								1			
	P31	0							1	0												-1											1		
	P32	0							-1		0	0	-1	-1								1						-1						1	
	P33	1											1	1																					1
Zj	-Cj	10	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	3	0	0	0	0	102	102	0	0	0	0	0	0	0

Figure 2

WE ARE AT THE OPTIMUM

Course Section 1 0 1 1 0 0 0 Monday Afternoon, Wednesday Morning
Course Section 2 0 1 0 1 0 0 Monday afternoon, Wednesday Afternoon

Note: Pattern 1 has floating periods
Pattern 2 has fixed periods

Of course, this problem has alternate optima and could have been solved by inspection because of its simplicity.

The reader should be aware that if all the C_{ij}'s are the same (i.e. all 1's) that the method finds solutions which satisfy all the constraints.

THE TURTLE CREEK SENIOR HIGH SCHOOL EXAMPLE

Applying this technique to a real high school made us more aware of the practical problems involved. In presenting this actual high school application I will not write down all of the original constraints on the solution nor will I present any detailed discussion of the computer program used in performing the computations.

First let me give a brief history of our experience with Turtle Creek. Through Dr. Snoke we were introduced to Dr. Cibik and the staff of the high school.

Giving a group of my graduate students an earlier paper of mine on the linear programming approach to scheduling, I assigned them to a term project of applying the theory to Turtle Creek. Providing guidance and tactical support as well as making necessary contacts for hardware (we used time on an IBM 7094 computer at Gulf Research and Development Corporation in Harmarville, Pennsylvania as well as on the university's equipment -- IBM 7070, 1401, and peripheral equipment) I monitored this group and a second group in the succeeding trimester as they collected data, wrote them into constraints, and used the computer for finding the solution.

The primary jobs which the students performed were as follows:

1. Gathering of basic data from Turtle Creek Senior High School

2. Writing the particular linear constraints for the master schedule generation problem at Turtle Creek using the general forms provided them,
3. Preparing these for input into the LP-90 computer program -- for this they wrote a Fortran program which would eliminate much manual punching,
4. Running the LP-90 program to find the solution to this particular problem.

THE SPECIFICS OF THE TURTLE CREEK SENIOR HIGH SCHOOL MASTER SCHEDULE GENERATION PROBLEM

Number of courses - 50

Number of course sections - 155 = J
(counting study halls)

Length of Schedule Cycle - 1 day - 7 periods = I

Number of Variables - 1168
((I x J) = (155 x 7) less a few which were unnecessary)

A typical variable

$X_{1,3}$ could take the value 0 or 1. If in the solution it has the value 1 this indicates that Algebra 1 is scheduled for 3rd period; if it has value 0 this means Algebra 1 is not scheduled 3rd period.

The variables are generally $X_{i,j}$ where i can be any number from 1 to 155 and j any number from 1 to 7.

Number of Constraints - 365

Most of the restrictions were of the type which prohibited the scheduling of a set of course sections together for one of three reasons:

1. Some set of students indicated they wanted to take these (to prevent conflicts in student schedules),
2. Some teacher or set of teachers had been assigned to teach this set of course sections (to prevent conflicts in instructor schedules),
3. Some room or set of rooms was of a specialized nature and had to be used by the set of course sections (to prevent conflicts in room schedules).

Note that these restrictions were all of the types five and seven as described in the general model above.

There were a few restrictions of other types also. An example would be a restriction on a particular set of course sections taught by the football coach who had to be free for practice during the 7th period.

I will not list all of the data on each restriction in this report. Further information is available if interested readers inquire via the mails.

The Solution

The master schedule generated by the computer, using the model outlined briefly above is presented below in Figure 3.

COURSE SECTIONS - COMPUTER SOLUTION - HUMAN SOLUTION

Figure 3 shows the list of course sections offered, the teacher assigned to teach it, and two different schedules;

1. the schedule generated by the linear programming technique and

2. The schedule actually being used -- one developed by the scheduler at Turtle Creek -- the former denoted by X, the latter by 0.

Course Sections			Time Periods							Teacher Assigned to Course Section
Code	Name		1	2	3	4	5	6	7	
1	Algebra I		0	X						Sleightholm
2	Algebra II	A			0		X			"
3		B				X	0			"
4	Trig.	C	X			0				"
5	Art I	A			0			X		Oakes (Barber)
6		B					X0			"
7	Art II	A	0			X				"
8	Art II & III	B			X	0				"
9	Art III	A	X					0		"
10	Study			X					0	"
11	Gen. Biology	A		0		X				Melocchi
12		B							X0	"
13		C	X0							Maffeo
14		D			X0					Maffeo
15	Biology	A		X		0				Melocchi
16		B			X		0			"
17		C	X					0		"
18	Bookkeeping	A					0	X		Collie
19	Bookkeeping	B				X		0		"
20	Chemistry	A	0		X					Shiring
21		B			0	X				"
22		C		X		0				"
23		D	X					0		"
24	Bus. Math.	A			X0					Collie
25		B		X		0				"
26	Mech. Draw		X		0					Aird
27	English III	A	0				X			Sarada
28		B		0				X		"
29		C			0	X				"
30		D			X			0		"
31		E		X					0	"
32		F			0			X		Cifra
33		G				0	X			"
34		H				X	0			"
35	English II	A	0	X						"
36		B	X	0						"
37		C			0			X		Roth (Burcaw)
38		D				0	X			"
39		E				X	0			"
40		F	0					X		Lazaro
41		G					X	0		"
42		H				X			0	"
43	English I	A		X0						Roth
44		B	X						0	"

Figure 3

Course Sections			Time Periods							Teacher Assigned to Course Section
Code	Name		1	2	3	4	5	6	7	
45	English I	C	X0							(Cenci)
46		D			0			X		"
47		E				X	0			"
48		F			X			0		"
49		G		X		0				"
50	Economics	A			X0					Lazaro
51		B	X			0				"
52		C	0				X			Michaels
53		D		0		X				"
54		E			X		0			"
55		F		X				0		"
56		G	X						0	"
57	Plane Geom.	A		0				X		Lewandowski
58		B				X0				"
59		C			X			0		"
60	Homemaking II					0		X		Kelly
61	History	A	0					X		Pantalone
62		B		0			X			"
63		C			0	X				"
64		D			X	0				"
65		E	X				0			"
66		F						X0		Patterson
67		G				X			0	"
68	Homemaking	A	0				X			Kelly
69		B		0		X				"
70		C		X	0					"
71	Latin I					X0				Seaman
72	II						X0			"
73	III							X0		"
74	Comm. Math.			0				X		Sleightholm
75	Office Pract.	A			0			X		Frankhauser
76		B	X			0				"
77		C			X				0	"
78	Physics	A	0		X					Lewis
79		B			0	X				"
80		C		X			0			"
81		D	X						0	"
82	Psychology	A	0					X		Lowman
83		B					X	0		"
84	World Cultures	A					X0			Detar
85		B						X0		"
86		C				X			0	"
87		D	0		X					Patterson
88		E		X0						"
89		F	X			0				"
90	Ess. Math				X0					Detar
91	Auto Shop	A	X0	X0	X0					Miller
92		B					X0	X0	X0	"
93	Steno I	A	X		0					Carthew
94		B					X0			"

Figure 3 (cont.)

Course Sections			Time Periods							Teacher Assigned to Course Section
Code	Name		1	2	3	4	5	6	7	
95	Study	A			X0					Seaman
96		B	X				0			Kelly
97		C		0	X					Cenci
98		D	0		X					Burcaw (Ruth)
99		E			X			0		Cifra
100	Typing II	A		0				X		Carthew
101		B			X	0				"
102	Typing I	A		X				0		"
103	Study					X				"
104	Typing I	B		0	X					Lowman
105	"	C		X		0				"
106	"	D	X						0	"
107	Study				0	X				"
108	Typing I	E	X0							Collie
109	Trig. & Solid	A	X	0						Detar
110		B		X		0				"
111	Transcription			X				0		Fankhauser
112	Steno II						X0			"
113	Spanish III					0		X		Barnyak
114	Spanish II	A					X	0		"
115		B				X			0	"
116	Spanish I	A	0		X					"
117		B		X0						"
118		C	X		0					"
119	Gen. Math	A		X	0					Lewandowski
120		B	X						0	"
121	Phys. Ed.	A	X							To be Named
122		B		X						"
123	Girls	C			X					"
124		D				X				"
125		E					X			"
126		F						X		"
127		G							X	"
128	Phys. Edu.	A	X							Ford
129	(Boys)	B		X						"
130		C			X					"
131		D				X				"
132		E					X			"
133		F						X		"
134		G							X	"
135	Band								X0	Rotilli
136	Chorus		X0							Blackwood
137	Sewing				X				0	Kelly
138	Machine Shop		X0	X0	X0					Watson
139	Elect. Shop						X0	X0	X0	Maffeo
140	Related Draw	A					X0			Watson (Machine Shop)
141		B		0						Aird (Electrical Shop)
142	Study		0			X				Fankhauser
143	Study						X		0	Collie
144	Drafting						X0	X0	X0	Aird
145	Journalism								X0	Seaman

Figure 3 (cont.)

Course Sections			Time Periods							Teacher Assigned to Course Section
Code	Name		1	2	3	4	5	6	7	
146	Study	A					X		0	Sleightholm
147		B					X0			Lewandowski
148		C			0			X		Melocchi
149		D	0					X		Aird
150		E				0		X		Michaels
151		F		X0						Lazaro
152		G		X				0		Pantalone
153		H					0		X	Patterson
154		I		0					X	Maffeo
155		J	X			0				Sarada

Figure 3 (cont.)

The reader may be quite surprised to realize that there are in fact a very large number of alternative solutions to this particular problem. One beauty of the linear programming approach is that we can easily find each of the alternative feasible solutions. In the schedule presented there is a very noticeable dearth of course sections scheduled for the 7th period. This is recognized and in future runs can be corrected by the addition of new restrictions such as one specifying the minimum number of course sections to be scheduled in a given time period.

ADVANTAGES OF THE L.P. APPROACH

The linear programming approach is a unique attack on the educational scheduling problem and has never before been carried out. Other people who have attempted this method have abandoned it as soon as they have seen the size of the problem, rather than taking a direct frontal attack on the issues size creates. This technique for tackling the problem is different from most other new approaches in that it is a strictly analytical method; once we formulate the problem we can prove mathematically that if a solution exists the technique will find it.

If we can carry out the computations it will be possible to find any existing solution and, if there are more, the means for finding all of them will be provided.

There are many reasons why we feel that this new approach is going to be better than a heuristic or a cut-and-try method. The first of these reasons is that with an analytical approach, if it is operational, it is much easier to arrive at an answer or to find out if an answer exists. Using the mathematical model we can deductively answer these kind of questions. This is in contrast to the heuristic approaches where one never knows whether a solution exists or not. You may find it, you may not, there is no way of telling.

In addition, this mathematical model of the system enables us to derive other very pertinent and useful information. This is information which administrators of educational institutions can put to very effective use. One of the most beneficial types of information is the so-called shadow prices which will provide means for evaluating the marginal returns of potential changes in physical plant, personnel resources, student demand, and administrative policies.

For example, if we have a constraint which says we can use no more than k classrooms and in a given solution we use all k, the pertinent shadow price will tell us the return, in whatever unit the criterion is measured, of adding one additional classroom of the right type to our facilities. This can be extremely useful to decision-makers in these kinds of institutions and, of course, there are analogous shadow prices for all of the other constraints in the system. Again, suppose an administrator sets a policy which says that no class will meet more than one hour a day (or 2, or 3, or k hours a day). We can evaluate the cost, in terms of the criterion we are using, of the existence or non-existence of any of these policies.

This again is something which has not been done in the past. Some of the heuristic programs attempt to do this but they are not rigorous and there is no assurance that the marginal cost is being accurately measured. This is a result of the fact that neither before nor after the constraint is included, can one be sure he has the optimum constrained solution. Another reason why we believe this approach is very useful is that, given the model of the particular system, (which is what we will have); when we

write down all the policy constraints, facility constraints, personnel constraints, etc. we can do research on the system itself to find out what would happen if ....
This is sometimes called simulation. Using simulation we can do some very basic research into the dynamics of the system and into the relationships which exist between various kinds of policy decisions, constraints, and the pertinent properties of the master schedules which can be generated from these systems. With the analytical approach we know what can be done with a given system. We can change the model of a system one way or another and determine the effect of such a change on the master schedules the system will now allow us to generate. This is an extremely powerful tool for studying the impact of various policy decisions; what does it mean, in terms of our master schedules, to make a policy decision of this type or that type or some other type? These are questions which have been largely unanswered up until now. We have essentially been doing the job of educational scheduling by the seat of our pants. We have guessed at many things and have used very limited information to find out the effects of various kinds of policies.

We have been discussing various criteria in the system but criteria for comparing feasible master schedules seldom exist and schedulers cannot decide what criterion is to be used to compare one master schedule with another. This is especially true if both master schedules to be compared satisfy all of the explicit constraints i.e. no faculty, room, or student conflicts. In the absence of such conflicts it is very difficult to evaluate a master schedule or even to compare two such master schedules. However, even in the absence of such criteria, without any criterion whatsoever, the L.P. approach does give us a means for generating feasible schedules. By feasible I mean schedules which satisfy all of the explicitly stated policy, personnel, facility, and student constraints. Not only does the L.P. method give us the tools to generate a feasible solution if one exists but it gives us the means to generate all the feasible solutions. We can count how many there are as we generate them. The number of feasible solutions is obviously dependent on the constraints. An area of great interest for further research would be a determination of the exact effects of each type of constraint. We could find out what determines the number of feasible solutions. Many people presently doing educational scheduling by hand or heuristic methods have formed the opinion that in most situations there are no feasible solutions because the large number of constraints make the solution space vanish. The proposed method will determine when this is true and when it is not without fail. In other words, this informal theory can be tested rigorously with empirical data. We can determine if it is valid in general and learn how to determine when it is valid in particular situations. These, I think, are all very important contributions.

RESEARCH NEEDS

The task of generating master time schedules for educational institutions has been formulated in a general linear model. Using real data from a local high school we have demonstrated the practical nature of this approach in a particular situation. We have encountered many problems in making the approach generally operational and hence have uncovered several areas of research needs. I will present these in the form of objectives of further research with ideas for achieving these goals.

Objectives of Future Research

1. To find analytical solutions to the task of sectioning students in such a way that the general objectives of the administration are satisfied and the number of alternative feasible* master schedules is maximized. Once this is done the generation of a master time schedule with linear programming (section assignments would be the source of some of the constraints) would automatically generate the student schedules also.
2. To formulate the remaining tasks in the general educational scheduling problem as linear models (for solution with linear programming) carefully indicating the mathematical means for writing generally used policies and constraints.

*See Glossary

3. To integrate all of the sub-task formulations into one large mathematical system and to develop a man-machine system for carrying out the complete scheduling job.
4. To make the linear programming approach to the generation of master time schedules operational for all cases which fit the general model. This would involve a large portion of any future efforts and would be of significant importance in producing useful and practical results from the research. The sub-goals -- the means to achieve this major goal -- would include:
    a. Finding techniques or approaches for surmounting or circumventing the problems resulting from the size of the linear programming problem. These constitute the major obstacles to the achievement of general operationality. (We are operational now for problems with no more than 1024 constraints.)
    b. Developing methods for translating the stated policies and the physical plant, faculty, and student demand constraints of a particular institution into the correct mathematical expressions necessary for input into the linear programming format.
    c. Investigating the development of criteria for educational scheduling and the related measurement problems.
    d. Investigating the potential usefulness of a special algorithm for solving the particular kind of linear programming problem presented by educational scheduling.

5. Given satisfactory achievement of some of the above objectives, to use the rigorous analytical model of the general educational scheduling problem for some very basic research into the relationships which exist between several pertinent variables in the educational scheduling process. Especially important are:

   a. The relationships between the number of different feasible master schedules and the several types of policy constraints which can be actuated at the option of the institution's administration. Some of these policy variables are:

      1'. the length of the minimum time block allowed in a schedule;
      2'. the allowance or prohibition of floating period schedules, floating instructors, or floating classrooms;
      3'. the length of minimum and maximum meeting times per schedule sub-cycle;
      4'. the length of the schedule cycle;
      5'. the length of the schedule sub-cycles;
      6'. the length of the schedule life - terms, semesters, quarters, etc;
      7'. the allowance or prohibition of non-consecutive meetings in a schedule sub-cycle;

   b. The relationships between the number of alternative 'best' schedules (given particular criteria and their actual numerical measures) and the variables listed above. Note that without a criterion the linear programming approach finds all schedules which satisfy all of the constraints written as linear inequalities. With a criterion it selects the best solutions among all the feasible solutions.

6. To study the 'cost' in terms of the various criteria developed, of the several common types of policy, physical plant, faculty, and student demand constraints.
7. To formulate and make operational this particular problem -- the assignment of classrooms to course sections, given a master schedule, so that faculty travel time to and from class is at a minimum.

PROPOSED PROCEDURES (AND ALTERNATIVES) FOR ACHIEVING THE STATED GOALS

This section deals with the procedures I would follow in the proposed future research and the alternative methods which I would keep in reserve as attempts were made to achieve the objectives enumerated in the preceding sections of this paper. Following Herbert A. Simon I will state these procedures as a set of sub-goals -- means for achieving the major objectives. Looking at the field in this way we can conceive of a hierarchy of means and ends.

The first objective was to find analytical solutions for the sectioning problem. My proposed procedure here would again make use of a linear,constrained model. I feel that this problem can be so structured that it too will be solvable using linear programming computational methods.

The second objective, in a real sense, included the first as a special case, but the first is such an important part that it seemed pertinent to mention it separately. This second objective was to formulate the remaining tasks in educational scheduling as linear programming problems, carefully indicating the mathematical means for writing the generally used policy, personnel, student, and facility constraints. Procedures for achieving this end need not be very elaborate nor need much effort be expended indicating what they might be. The first step would involve a comprehensive survey of educational institutions to discover operating constraints being used in the field. Next, a close look at each of the tasks with the aim of building a linear mathematical model, would precede the actual model construction. We want to build the models for the general case including all possible kinds of constraints, for all types of institutions. We would, of course, try to insure that they are all linear since linear programming is the field in which we would like to do our work. This is because in the field of mathematical programming linear programming is the most highly developed and the most operational. This work, of course, would be an initial step towards making this system practical and useful, one of our later goals.

Thirdly, I propose to integrate all of the subtask formulations into one large mathematical system. I would take each of the tasks and its constraints as written in the second part of this research and tie them all together in one system. The tie-ins, of course, would be mathematical. For example, only a given number of students can be assigned to a classroom because of the size of the classroom. Hence, students and classrooms are closely related. Only as many classrooms can be used, in a given period, as there are instructors. This section of the research would require the writing of those additional constraints which over-lap, in terms of the sub-tasks of the educational scheduling problem. The survey results would be very useful here, too.

The fourth objective, as stated in the earlier section, is to make the linear programming approach to educational scheduling operational, that is, useful and practical. This would involve major portions of any future research effort and would, in my opinion, be the single most important goal of that work. This is such a major objective and is so critical that I outlined the sub-goals for achieving it in the section above on objectives. The first sub-goal is: finding techniques or approaches which will surmount or circumvent the problems resulting from size. A great deal of emphasis should be placed on the fact that the greatest hurdle in making the linear programming approach operational for the general case is this problem of size. When we state the master time schedule generating problem in a linear programming format with variables defined and all constraints written mathematically in terms of relations between these variables, we get such huge problems that we have had little hope of computationally solving them up until now. We can write them down and we know that solutions exist but there are no practical means for finding the answers to most of these (we can generally solve those with less than 1024 constraints) at present. This is true for any medium to large size scheduling problem. Many people in the past have given up when they realized this fact but since we know what the nature of the problem is, it would seem that a frontal attack on this problem could pay dividends. The procedures or sub-goals for getting around it take many forms.

One approach would be to sub-divide the problem in such a way that we do not lose anything from the solution but save ourselves some of the headaches of size since we can solve the parts sequentially and thus the whole problem. This approach is sometimes called a partitioning of the problem. There has been some recent literature in mathematical programming on partitioning but much work is required to adapt it operationally to this specific class of linear programming problems.

The second proposed approach for getting around this size problem is to develop procedures for detecting and eliminating redundant or unnecessary constraints. The elimination of any unnecessary restrictions will certainly save space in computer storage and will cut down the effective size of the problem. Redundant restrictions write explicitly what other constraints implicitly guarantee and hence are unnecessary and burdensome. This work would be an important source of size reduction.

The third proposed approach is to take part of the system, as many constraints as we can store, and solve it using all the variables (we are generally not constrained as to the number of variables in our better linear programming computer codes, the number of constraints is usually the limiting dimension of the problem). There will be many alternative solutions -- this set of solutions would be listed. These would be the alternative schedules which satisfy this first subset of restrictions. Next we take another (different) set of constraints, find the set of solutions satisfying this set of constraints and so on until we have used all the constraints. We know from the theory of sets that the solutions in intersection of all these sets are the solution(s) to the total problem.

A fourth procedure we might contemplate using is one which saves the power of the linear programming computational algorithm but gets around some of the size problems. Although it may cost us in the form of additional computation time we need to investigate this to find out if we can gain operationality at an acceptable cost. For example, we think it is worth trying to solve the system with as many constraints as possible, choosing these carefully, and from the set of alternative feasible solutions with these constraints, sequentially filter these through the rest of the constraints directly to select those which satisfy the whole set of restrictions on the system.

I have some other alternative modes of attack in mind but I will not make them explicit here since I believe the four mentioned could provide us with a great deal of work and would generate much new knowledge, not only about linear programming, but also about educational scheduling.

The next sub-goal is; developing methods for translating the administrative policies, plant, personnel, and student constraints into mathematical expressions for insertion into the linear programming system. It is very often the case in educational institutions that the policies of the institution, facility and human resource constraints, and student demand constraints are either not even formally recognized or are described verbally. They must be stated in mathematical symbols in order that the linear programming approach can be operational. We have seen in the section on the general linear model that certain types of policy constraints which are very easy to state in words generally require large numbers of mathematical inequalities which are the inputs for the L.P. mode of solution. To write all of these down by hand would be an extremely time-consuming task. It is very important and very necessary that programs and techniques be generated which will allow the users of our system to quickly translate the various policies and other types of constraints into the necessary mathematical expressions for inputting into the linear programming program.

Along the same vein but a means for getting around the size problem would be researching the problem of combining constraints. The familiar conflict matrix is presently used for generating constraints. It tells us about pairs of course sections which cannot be scheduled together. A matrix is a rectangular two dimensional array of rows and columns. I believe that an investigation into using larger dimensional arrays would allow us to be more efficient in the generation of restrictions. These would also, I feel, help to cut down the number of different constraints and hence the crucial size dimension of the problem.

The next sub-goal of the major goal -- making the linear programming approach generally operational -- is, investigating and developing criteria and their measures, in educational scheduling. The means I would propose for handling this part of future research include; (1) a survey of presently used formal and informal criteria in current practice (I expect the informal type is more predominant) (2) the testing of some alternatives to be developed by the research.
I would expect that such research would make it possible to design techniques for surveying a faculty and student body to determine, for example, what the 'relative desirability' of using the different available time periods for the different course sections is. This work would be a substantial time consumer but would, in my opinion, solve some measurement problems in an operational manner and would allow educational administrators to quantitatively compare feasible alternative master time schedules. We could, once criteria and means

for measuring them are developed, compare alternative ways of handling the numerous tasks in educational scheduling.

Also along this line, would be work with the problems of multiple criteria. Especially pertinent here would be the development of methods for making choices among alternative schedules when the choices must be made on the basis of many criteria. Procedures which might be investigated here would include; (1) transforming the measures of each of the criteria to some common measure, (2) using all but one of the criteria as constraints, setting absolute limits, either high or low above which or below which we will not allow our variables to go, and try to maximize (or minimize) the remaining criterion subject to the above mentioned restrictions. This is similar to what we do in the linear programming approach. It is significant, I feel, to note that cut-and-try schedules state as one of their goals the minimization off *student conflicts* whereas in the L.P. approach we use *constraints* which explicitly say there will be *no* conflicts.

The next sub-goal was -- investigating the potential usefulness of special algorithms for solving the particular kind of L.P. Model presented by educational scheduling. The structure of the linear programming problem generated by the master time schedule problem is of a type generally known in the field of mathematical programming as a modified transportation type. However, it does have its own special characteristics and I feel that research into the special problem of finding a particular type of computational rule which will be more efficient than anything now known for this special kind of problem may be very well worth its cost. I do feel that there is a good chance that such a special algorithm could be developed.

Major goal five was, as stated: given satisfactory achievement of the earlier objectives, to use the rigorous analytical model for some very basic research into the relationships which exist between several pertinent variables in the educational scheduling process. The techniques which would be used here are standard in the field of management science. They are often called by the generic name of 'simulation'. Once we have the mathematical formulation of the model in the context described above we can simulate different sets of policies and constraints on the system. Given one such set we can solve the problem for the feasible master schedules and we can count how many exist. We can then change one or some of the constraints, run the simulation again, and compare the numbers of feasible schedules in the two situations. We could do this and discover the relationship which exists between the various policy decision variables and attributes of the solutions to the system. We could also discover the relationships which exist between the various policies and the value of the criterion function at the optimum (once the criteria and their measures are developed). Obviously, there are many quantitative relationships which we could explore using this technique. Those I mention here are a small sample of the very large set of things which we could learn from this model.

Major goal six was to study the cost of the several common types of policy, physical plant, faculty, and student demand constraints in terms of the various criteria developed. The procedure followed here would be very much the same as followed in attaining goal five. We would use the model to simulate various conditions and measure the cost of changing these kinds of constraints.

Major goal seven could be accomplished in quite a straight-forward manner using the linear programming transportation type model with classrooms being 'delivered' to course offerings in such a way that some travel-distance criterion function is minimized.

There is one more specific means for achieving some of the major objectives. Such a sub-goal would be the development of techniques for measuring student demand for the course offerings, and to work towards the discovery of methods which would enable us to write demand functions for courses. These would be analogous to the demand functions in classical economics. Techniques used here would again involve the collection of empirical data from typical institutions of varying types. The work would be statistical and mathematical yet, one would hope, practical. We would hope to develop tools which are general in nature and yet easily adaptable to specific situations.

GLOSSARY

Schedule Cycle - In this paper I use this term to indicate the length of time over which a schedule does not repeat itself. Bush, Caffrey, Oakford, and Allen used the term Schedule Module for this same concept.[*]

J - The symbol used to denote the length of the schedule cycle in units of usable time periods. Hence it also represents the total number of usable time units per schedule cycle.

j - A general subscript to denote *any* time period - the jth-.

I - The number of course sections to be scheduled for an institution.

i - A general subscript used to denote *any* particular course section.

Time Block, Time Unit, Time Period - Terminology used to indicate time intervals from the schedule cycle. In our formulation we are required to use a unit at least as small as the smallest time intervals which are allowed in a schedule pattern. We have J of these, all equal in length, in a schedule cycle. We are also required to use only integral multiples of this size interval in the schedule patterns. This should not be a hindrance however, since we *can* use a very *small* interval if the need arises.

Minimum Time Block - This is the smallest interval of time which the institutional managers will allow any schedule pattern to use.

Subcycles, Sub-subcycles, etc. - We use this terminology to indicate various cycles of time within the schedule cycle; most often, but not in all cases, these subcycles will be functions of natural and calendar phenomena. For example, if the schedule cycle were a month, a subcycle could be a calendar week, and a sub-subcycle could be a calender day.

Because of the need for even multiples of minor cycles in all major cycles we can make use of a common 'trick'. In the example, we could assume all months are five full calender weeks in length. Then we would write constraints in our model which tell us never to use the excess time periods. Much research is needed on the subject of the influence of various constraints on, and choices of, minor cycles.

---

[*] Bush, Robert N., John G. Caffrey, Robert V. Oakford, Wright W. Allen; "Using Machines to Make the High-School Schedule; *The School Review*, Vol. 69 (Spring, 1961) pp. 48-59.

n - A general notational symbol used here to indicate the number of sub-cycles of time in a schedule cycle. We made the further assumption (an arbitrary choice) that n is an integer.

J/n - General, symbolic notation to indicate the number of time units in a schedule subcycle. We earlier assumed 1) that n was an integer and 2) that J was an even multiple of n.

Feasible Schedule Pattern - Any assignment of time units to a schedule pattern, from the list of J available, which does not violate any policies or requirements of the administration.

Linear Programming - A computational device for finding the extreme point or points of a particular class of functions constrained by a particular type of restraining relations. The functions and the constraining relations must be linear in the variables, hence the name, Linear Programming. The reader may find additional information about this subject in any of the books or articles noted in Appendix I.

Algorithm - A special word used in mathematics. It is the name used to denote any of a large class of computational rules used to find solutions to mathematical problems.

Schedule Pattern - The term used in this paper to denote a time schedule. If we think of a series of J slots, each representing a time unit, then any assignment of these units will be called a schedule pattern.

A way of handling the educational scheduling task would be to generate the schedule patterns and then to assign the course section offerings to these.

Schedule Life - The length of time during which a master schedule is unchanged. Typical choices have been semesters, terms, etc. This is a policy variable and its effect on schedule flexibility and other criteria is an underdeveloped research area. Bush, Caffrey, et. al., use the term schedule cycle for this same time interval.

Fixed Period Schedule Pattern - A schedule pattern which never uses time intervals from different levels of any two schedule subcycles. A schedule pattern such as Monday, Wednesday, and Friday at x o'clock is a fixed period schedule pattern.

Floating Period Schedule Pattern - A schedule pattern which is not the fixed period type. An example would be Monday at 9, Wednesday at 10, and Friday at 11.

Unconstrained Period Schedule Pattern - A schedule pattern which can be either fixed, floating or mixed.

Flexibility - As used in this paper this term means the number of alternative feasible (if no criteria exist) or best (if criteria do exist) assignments subject to a given set of policies. This number can be used as a direct measure of flexibility, the larger it is the more flexible the system.

Independent Schedule Groups - Those groups of students which have no schedule interactions. Hence, they can be treated independently. Example - Freshman and Seniors are two independent schedule groups in most educational institutions.

Offer List - The list of course sections to be offered during a particular schedule life.

Constraint, Restraint, Constraining or Restraining Relations - These are mathematical terms used to denote the expressions which mathematically represent the policies of the administration with regards to the scheduling jobs. These policies, in effect, state what is or is not allowed in terms of solutions to the various assignment problems. The symbolic notation merely restates the same policies in a more precise language.

Cij - This is a general symbol used to denote the criterion measure for using the jth time period in the ith schedule pattern. It may be a measure of desirability, cost, revenue, etc.

If we deal with the assignment of time units directly to course sections the Cij represent the criterion measures for using the jth time interval for the ith course section offering.

Xij - A general symbol used to denote whether or not the jth time unit is used for the ith course section offering. We constrain all the variables Xij to the values 0 or 1, 0 denoting that the jth period is not used in the ith pattern and 1 denoting that it is used.

PROCEDURE, ANALYSES OF DATA, AND FINDINGS

HEURISTIC SCHEDULER

A HEURISTIC APPROACH TO THE GENERATION OF A MASTER SCHEDULE AND ASSIGNMENT OF STUDENTS

BY

CHARLES E. DONAGHEY
OSMAN S. OZKUL

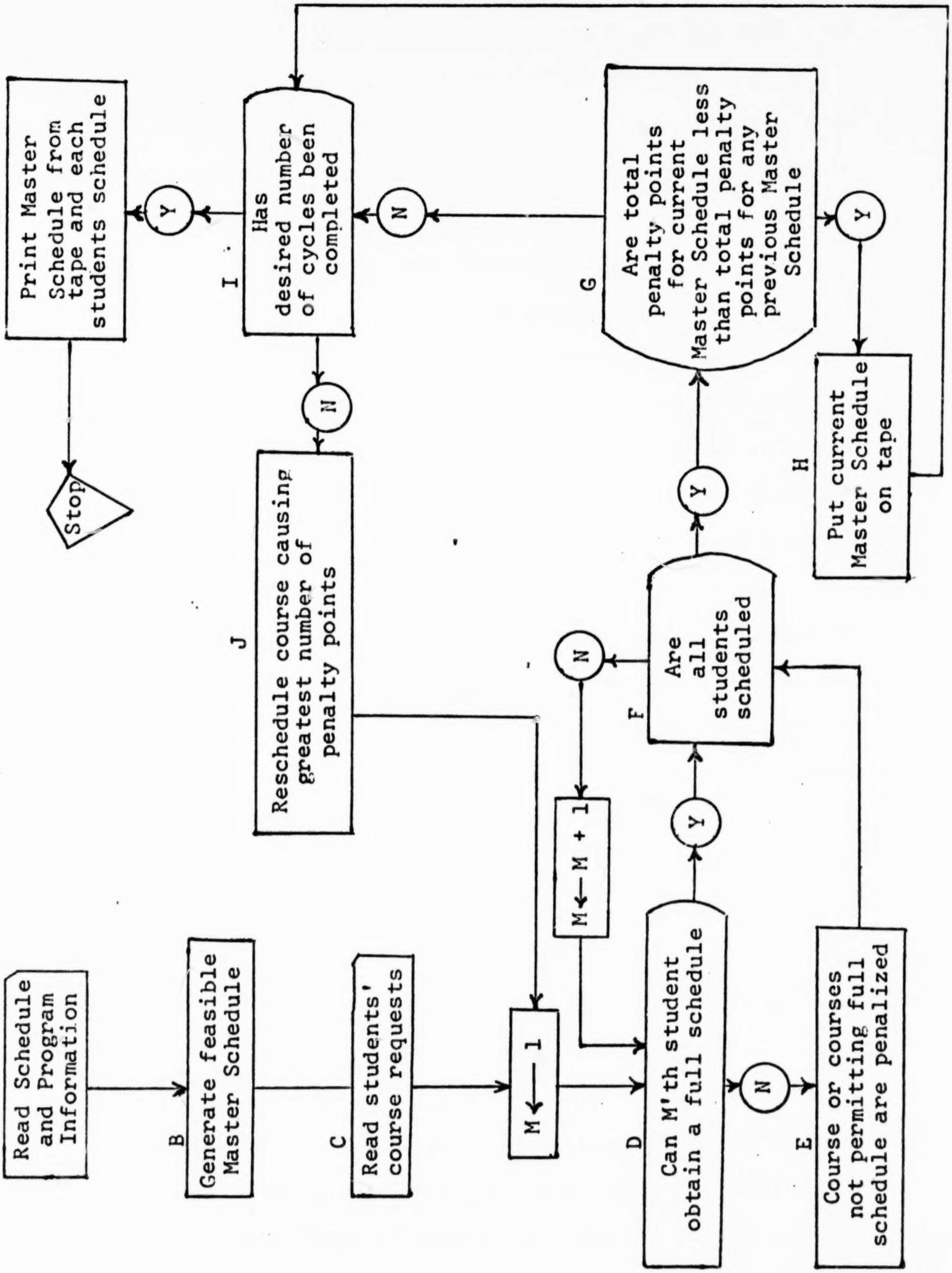
INTRODUCTION

The idea underlying this study was that if a master schedule could be developed that contains no faculty or room conflicts, by scheduling all students to this original master schedule, determining the course that caused the greatest number of student conflicts, rescheduling this course, and repeating the cycle, we would ultimately come up with a master schedule in which the student conflicts were reduced to a minimum. Such a program has been developed. The success with this program has so far been encouraging.

This report will describe our early work, the results of experiments with university and high school data, the improvements that were made to the initial programs, and the future work we feel should be done in this area.

INITIAL WORK

In early 1963 a paper was prepared that outlined our first ideas on a heuristic approach to the problem of generating a master schedule and assigning students to this schedule. This paper is reproduced on the following pages to outline our original thinking. On the next page is a general flow chart of this program; following is an explanation of each step of the chart.


A
Read Schedule and Program Information
B
Generate feasible Master Schedule
C
Read students' course requests
M ← 1
D
Can M'th student obtain a full schedule
N
E
Course or courses not permitting full schedule are penalized
Y
F
Are all students scheduled
N
M ← M + 1
Y
G
Are total penalty points for current Master Schedule less than total penalty points for any previous Master Schedule
Y
H
Put current Master Schedule on tape
N
I
Has desired number of cycles been completed
N
J
Reschedule course causing greatest number of penalty points
Y
K
Print Master Schedule from tape and each students schedule
Stop

A. Read Schedule and Program Information

In this operation the machine is fed the information on the courses that are to be offered, the number of sections of each course, the instructors for the courses, the maximum number of students allowed in each section, and the number of hours per week each course must meet. If a course must meet in a specific room, such as a typing class, this is also fed into the machine. The computer is also told of the courses that must meet at already decided upon times. For example, if the band director is at the junior high school for only the first period each day, band must be scheduled at that time.

To help the school administrator decide how many sections of a course should be offered, another program may be used, with the students course request cards as input. The output from this program is a tally of the number of times each course was requested.

The number of cycles the machine is to run through is also brought in at this point. Other program information, such as the number of courses that constitute a full schedule for a student, may be designated.

B. Generate a Feasible Master Schedule

A feasible master schedule is considered here as a schedule where there are no faculty or room conflicts. In this program specific student requests have not been taken into account. The inputs to the program are the courses to be offered, the number of sections of each course, faculty and room specifications, and the number of hours that these courses meet in a week. Provisions are made to take care of the courses that meet at consecutive hours. These should be indicated in the input data.

Conventional time patterns are built into the program for courses that meet one day a week, through the courses that meet five days a week. Since laboratories are treated separately from their lecture sections, it is felt that the large majority of courses in high schools or colleges will not meet more than 5 times a week and thus be within the limitations of the program. For schools that meet six days a week additional time patterns will have to be built into the program. A few example time patterns incorporated in the program are the following: (1) Monday, Wednesday, Friday first hour, and Monday, Wednesday, Friday fifth hour are two of the time patterns for courses that meet three days a week; (2) Tuesday and Thursday sixth hours is one of the time patterns for courses that meet two days a week.

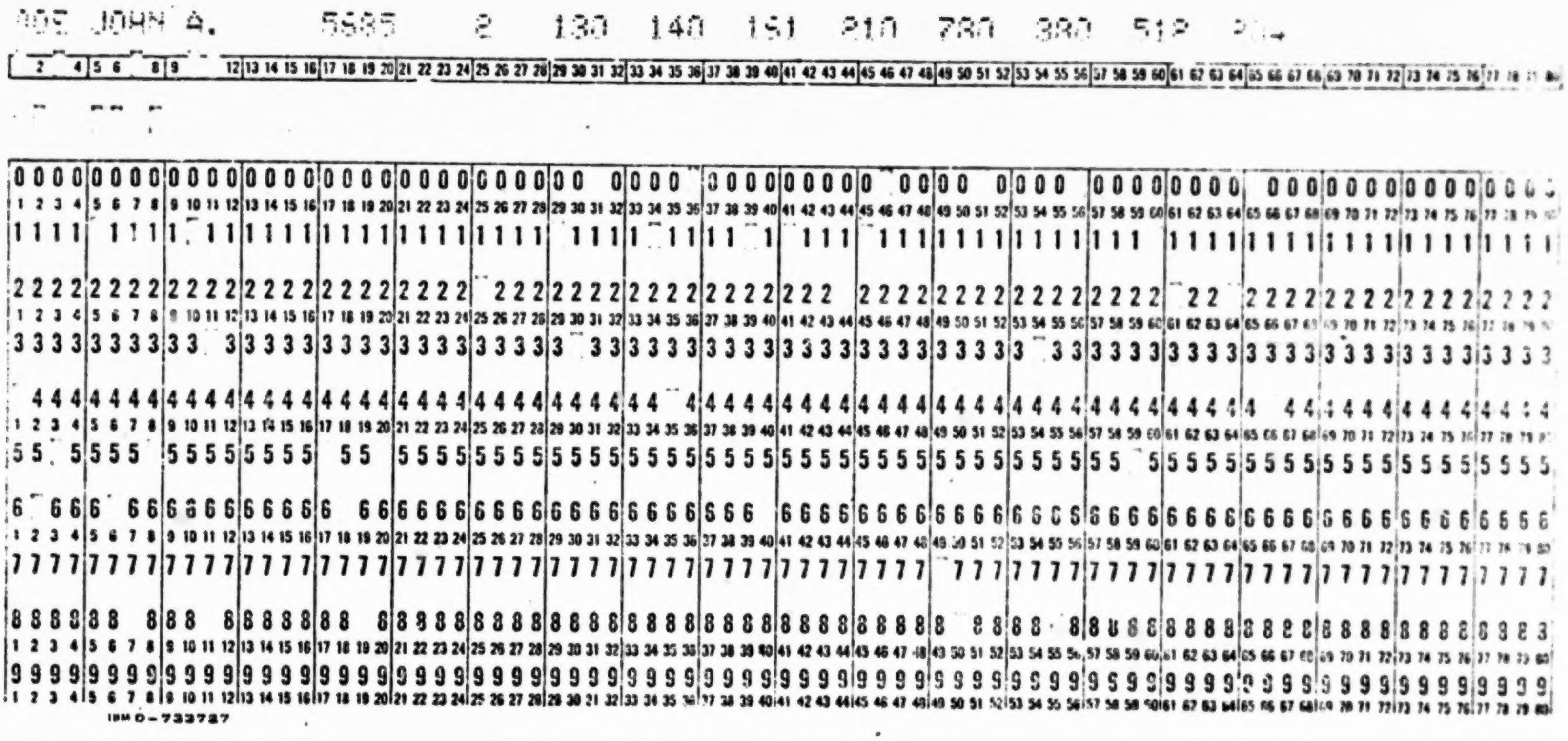
Each time a course is randomly assigned to an appropriate time pattern, the computer checks first to see that it will not cause a faculty or room conflict. If there is a conflict another random assignment is made and this cycle is repeated until a time pattern without conflicts is found, or the specified number of trials is exhausted. If no time pattern without conflicts is found within the specified number of trials, the computer prints out a message indicating the cause of the trouble and goes to the next course. These unscheduled courses can be assigned manually by changing their faculty or room specifications if no time pattern without conflicts is available.

The number of attempts the computer is to make in scheduling a course is an input parameter. It has been found if the machine does not find an available time pattern in 25 attempts one probably does not exist.

C. Read Students' Course Requests

The student's name, his student number (if used), and the courses the student would like to take are fed into the machine. A student can request any number of courses.

A status number for each student can also be brought into the machine at this point. This feature can be utilized or not. If a school administrator has a group of students he feels must be scheduled to a full load, and another group to which a full load is not as important, he can assign a high status number to the first group and a low status number to the second. The machine will endeavor to resolve the conflicts in the first group to a greater extent than in the second. How this is done will be shown later. If all students are to be treated equally, they should all be assigned a status value of one. Below is a typical student request card.



Columns 1-15, Student's Name
Columns 16-20, Student Number
Column 25, Status
Columns 29-80, Course Requests

D. Can M'th Student Obtain a Full Schedule

E. Course or Courses Not Permitting Full Schedule are Penalized

F. Are all Students Scheduled

The machine will take the first student's first request and see if it is the same as the first course in the master schedule list. If it is not, it will go on down the master schedule list until it matches the request with the course in the master schedule list. The first student will then be assigned to this course and these periods will be blocked out in the student's schedule. The student's second request will then be considered and matched with the course in the master schedule list. If the meeting periods for this course have not been blocked out previously in the student's schedule, he will be scheduled to this course.

Each time a student is scheduled to a course a tally for the course is increased by one. Before a student is scheduled to a course the tally for the course is checked with the maximum number allowed in the course, and the course is closed when the tally reaches the maximum number allowed in the course. When a student requests a course that cannot be scheduled for him due to the meeting times of the course conflicting with blocked out periods in his schedule or the course previously being filled, the computer will look for an additional section of the course that has room and will fit the student's schedule. If a student's request cannot be scheduled due to a conflict, and there are no additional sections of the course to which he can be scheduled, the course is penalized. Each course has a penalty tally which is increased each time the course causes a conflict. The amount this penalty tally is increased depends upon the status of the student. If a student with a status of eight has a conflict, the course causing the conflict will be penalized eight points. If it was a student with a status of one, the conflict-causing course would be penalized one point. It is also possible in the program to have the machine penalize courses more if they are higher priority courses in the student's request list. For example, a conflict for the student's second request would penalize the course more than a conflict for the fifth request. The assignment of these penalty points requires further research; however, the weighing of penalty points can be ignored in the program by setting the status of each student to one and informing the machine that a conflict is worth only one penalty point regardless of the priority of the request. When a student has a full schedule, no more requests are considered and the machine goes to the next student's requests. The number of courses that determines a full schedule has been previously read into the machine or can be shown individually for each student on his request card.

G. Are Total Penalty Points for Current Master Schedule Less Than Total Penalty Points for Any Previous Master Schedule

H. Put Current Master Schedule on Tape

I. Has Desired Number of Cycles Been Completed

After each student has been scheduled or an attempt made to give each a full schedule, the machine will total all the penalty points for all of the courses. On the first cycle, when the students are scheduled to the random master schedule, the schedule is put on tape. The master schedule is then adjusted and all the students are then scheduled with the adjusted master schedule. The total penalty points for the adjusted master schedule are then compared to the total penalty points of the schedule on tape, and if the adjusted schedule has less penalty points, it replaces the schedule on tape. The schedule is again adjusted and the cycle begun again. In this way the best master schedule will be retained on tape. The machine will go through this cycle any number of specified times.

J. Reschedule Course Causing Greatest Number of Penalty Points

The course which has the greatest number of penalty points is rescheduled each time before the cycle is begun again. For example, if the worst course meets Monday through Friday at 8:00 A.M., the machine will reschedule it for Monday through Friday at 9:00 A.M. and go back and reschedule all the students to the master schedule with this adjustment. The machine will not reschedule a course to a time that will cause either a faculty or a room conflict. If a course cannot be rescheduled, the machine will determine this, print out a message to this effect, and then reschedule the next worst course. This could occur with a class such as typing that must meet in the typing room. If the typing room is busy the entire day, the machine will not be able to reschedule a class in that room. Courses that have consecutive meeting hours will continue to have that sequence even after rescheduling. For example, if the course to be rescheduled meets the last two hours Monday, after rescheduling it will not meet one hour on Monday and one hour on Tuesday--both hours will still be consecutive on one day.

K. Print Master Schedule From Tape and Each Student's Schedule

After each cycle the machine will print information on the master schedule it has just considered. It will print the total penalty points for the schedule, the worst course, the penalty points assessed this course, and the number of students who got ten, nine, eight, seven, six, five, four, three, two, or one course with this schedule. After it has gone through the specified number of cycles, it will print the best master schedule it developed and will show each student's weekly schedule with this master schedule. The student's weekly schedule can also be punched out and this data used as input for another program which will prepare class rosters for all the courses.

INITIAL RESULTS WITH UNIVERSITY DATA

This system was tested with sample data from the Engineering School of the University of Pittsburgh. The results were encouraging. With a sample of 40 course section offerings and 180 students (each student requesting an average of 5-1/2 courses), the number of unscheduled course requests was reduced 40%, by making 20 cycles and adjustments to the Master Schedule. Each cycle took approximately 4-1/2 minutes of 7070 computer time, so with 90 minutes of total computer time we generated a master schedule that had 40% less conflicts than the original feasible master schedule. There did not seem to be any leveling off of the improvement, and we believe the master schedule would have continued to be improved if the computer were allowed to run through more cycles. We did not attempt this because of budget limitations. With a machine such as the 7090, in the same 90 minutes of computer time, at least 60 cycles and adjustments to the master schedule could have been made.

Figure 1 shows output from a sample run of 28 of the students in the Engineering School test. With the randomly generated master schedule there were 258 conflicts and course 190700 caused 24 of these conflicts. This is the course the machine re-scheduled for the next cycle. The information on the first schedule can be found on the line headed "Schedule No. 1". After course 190700 was re-scheduled and the students assigned to the revised master schedule, there were 222 conflicts and 40700 was the worst course. The machine was set to go through 30 of these cycles in this experiment. It can be seen in this case that schedule 25 gave the minimum conflicts (144) and this schedule is the one the machine retains on tape. Schedule 30 also gave only 144 conflicts but since it was no better than the previous best, it is not retained.

The machine then prints out the best master schedule. On the next page the beginning of this best master schedule is shown. The first line of the schedule is reporduced here:

```
171800   112   0   1   2   17   18   33
```

This means that course 171800 is taught by faculty member 112, there were 0 people assigned to this course and it meets periods 1, 2, 17, 18 and 33 during the 40 hour week. Since we were dealing with a small sample of students most of the courses do not have any students assigned to them.

In this example the number of conflicts decreased from 258 to 144, a reduction of 44%.

It is to be noted, however, with University data, that a faculty member is seldom required to teach more than 15 hours out of a 40 hour week. The machine, therefore, was able to re-schedule a course with a minimum of difficulty from faculty conflicts. Rooms are also not as constrained as in high schools.

```
DONAGHEY                                        36   25RINDUE    395       FRI., APR. 27, 1962
TAPE 12 USED AS SCRATCH

EXECUTE FORTRAN

SCHEDULE NO.  1 POINTS         258WORST COURSE 190700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      6   NO. WITH 3 COR.      7   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  2 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  3 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  4 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  5 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  6 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  7 POINTS         222WORST COURSE  40700 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      8   NO. WITH 3 COR.      5   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  8 POINTS         242WORST COURSE  42100 POINTS          24
NO. WITH 5 COR.       4   NO. WITH 4 COR.      6   NO. WITH 3 COR.      7   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO.  9 POINTS         212WORST COURSE  40800 POINTS          20
NO. WITH 5 COR.       5   NO. WITH 4 COR.      5   NO. WITH 3 COR.      7   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 10 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 11 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 12 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 13 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 14 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 15 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 16 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 17 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 18 POINTS         168WORST COURSE  10800 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 19 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 20 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 21 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 22 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 23 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 24 POINTS         152WORST COURSE 110200 POINTS          16
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 25 POINTS         144WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 26 POINTS         150WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       4   NO. WITH 4 COR.      9   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 27 POINTS         150WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       4   NO. WITH 4 COR.      9   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 28 POINTS         150WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       4   NO. WITH 4 COR.      9   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 29 POINTS         150WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       4   NO. WITH 4 COR.      9   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
SCHEDULE NO. 30 POINTS         144WORST COURSE  41300 POINTS          12
NO. WITH 5 COR.       5   NO. WITH 4 COR.      8   NO. WITH 3 COR.      4   NO. WITH 2 COR.      4   NO. WITH 1 COR.      7
```

Figure 1

```
THE BEST SCHEDULE OF THE ABOVE IS SHOWN BELOW
IT IS NO.   25
171800      112      0      1      2     17     18     33

181000      113      0      9     10     25     26      0

190500      114      3     20     36      0      0      0

190600      115      0     10     26      0      0      0

190700      116      2     39     15      0      0      0

191200      117      1      5     21     37      0      0

100100       63      0      3      0      0      0      0

100200       64      0      6      7      8      0      0

100300       65      0     19      0      0      0      0

100400       66      0     22     23     24      0      0

100500       67      0     29      0      0      0      0

100600       68      1     13     14     15      0      0

201100      118      0     11     27      0      0      0

201200      119      0     30      0      0      0      0

201300      120      0      2     10     18      0      0

201301      122      0      7     15     23      0      0

201400      123      1      5     21     37      0      0

201600      124      0      1      9     17     25     33

201700      125      0      3      0      0      0      0

201900      126      1      2     18     34      0      0

201901      126      0      5     21     37      0      0

202100      127      0      3     11     19     27     35

202600      128      0      2     18     34      0      0

202900      129      1      5     21     37      0      0

203000      130      0      4     20     36      0      0

203000      130      0      4     20     36      0      0

203200      131      0      4     20     36      0      0

203400      132      1      2      0      0      0      0

203700      133      0      4     20      0      0      0

204000      134      0      2     10     18     26     34

204100      135      0     13     29      0      0      0

204200      136      0      2     18     34      0      0
```

Figure 1 (cont.)

Figure 1 (cont.)

204201	137	0	5	21	37	0	0
204300	138	0	6	22	38	0	0
204600	139	0	1	17	33	0	0
204700	140	1	4	20	0	0	0
204800	141	1	4	12	20	0	0
205000	142	0	4	12	20	28	36
205200	143	0	1	17	33	0	0
205500	144	0	4	20	36	0	0
205501	145	0	7	23	39	0	0
205800	146	0	5	21	37	0	0
205900	147	0	1	17	33	0	0
205901	148	1	2	18	34	0	0
205902	149	0	5	21	37	0	0
205903	150	0	13	14	29	30	0
205904	151	0	6	22	38	0	0
205905	152	0	7	23	39	0	0
205906	153	0	1	17	33	0	0
205907	154	0	2	18	34	0	0
205908	155	0	3	19	35	0	0
205909	156	0	4	20	36	0	0
205910	157	0	3	19	35	0	0
206400	158	0	2	18	34	0	0
10100	1	0	9	0	0	0	0
10500	2	1	11	27	0	0	0
10600	3	1	1	2	0	0	0
10700	4	1	33	0	0	0	0
10800	5	1	18	19	20	34	35
11000	6	0	3	19	35	0	0
11200	7	0	4	20	36	0	0
110100	88	1	4	20	36	0	0
110101	89	0	5	21	37	0	0
110200	90	1	34	10	26	0	0
110300	91	0	2	10	18	34	0

Figure 1 (cont.)

110500	92	0	13	14	15	0	0
110501	93	0	25	26	27	0	0
110600	94	1	2	18	34	0	0
110601	95	0	6	22	38	0	0
20200	18	0	2	18	34	0	0
20300	19	0	30	31	32	0	0
21100	20	1	40	0	0	0	0
21200	20	1	25	26	27	0	0
21500	20	0	5	21	37	0	0
21600	20	0	2	3	4	0	0
21700	21	3	16	31	32	0	0
21800	21	3	37	38	39	0	0
21900	22	1	2	18	34	0	0
22000	23	0	3	19	35	0	0
22100	20	1	10	11	12	13	14
22400	20	2	10	11	12	13	14
22500	18	0	15	39	0	0	0
22600	24	0	25	26	27	0	0
22700	18	2	6	38	0	0	0
22800	25	2	10	11	12	0	0
23600	26	0	4	20	0	0	0
23700	27	0	6	7	8	0	0
23800	18	1	36	0	0	0	0
23900	18	1	22	23	24	0	0
30100	8	3	25	26	27	28	29
30600	9	0	2	18	0	0	0
30700	10	0	34	35	36	0	0
30800	11	0	3	19	0	0	0
30900	12	0	10	11	12	0	0
31200	13	0	9	0	0	0	0
31300	14	0	25	26	27	0	0
31700	15	0	6	30	0	0	0
31800	16	0	14	15	16	0	0

Figure 1 (cont.)

32700	17	0	13	29	0	0	0
131500	100	0	33	0	0	0	0
131700	101	0	1	17	33	0	0
131800	102	0	1	17	0	0	0
40200	28	2	2	18	34	0	0
40201	29	0	1	17	33	0	0
40300	30	1	1	17	0	0	0
40301	31	0	2	18	0	0	0
40400	32	1	10	11	12	0	0
40401	33	0	14	15	16	0	0
40402	34	0	25	26	27	0	0
40403	35	0	30	31	32	0	0
40500	36	2	3	19	35	0	0
40600	37	2	10	11	12	0	0
40601	38	0	30	31	32	0	0
40700	39	3	38	6	14	30	0
40800	40	0	6	7	8	0	0
40801	41	3	22	23	24	0	0
40900	42	2	5	21	37	0	0
41000	43	2	6	7	8	0	0
41001	44	0	22	23	24	0	0
41100	45	1	6	22	38	0	0
41200	46	0	36	12	28	0	0
41300	47	1	2	10	18	34	0
41400	48	1	14	15	16	0	0
41401	49	1	25	26	27	0	0
41500	50	0	1	9	17	33	0
41600	51	0	25	26	27	0	0
41601	52	0	30	31	32	0	0
41700	53	1	7	23	31	39	0
41800	54	1	14	15	16	0	0
41801	55	0	25	26	27	0	0
41900	56	0	7	23	39	0	0

Figure 3 (cont.)

42000	57	0	14	15	16	0	0
42001	58	0	30	31	32	0	0
42100	59	1	5	6	21	22	0
42200	60	1	3	19	27	35	0
42300	61	1	2	18	34	0	0
42400	62	0	9	25	0	0	0
140300	103	1	4	20	0	0	0
140600	104	1	5	21	0	0	0
51500	69	1	5	21	37	0	0
51501	69	0	2	18	34	0	0
51800	69	0	5	6	7	0	0
52000	70	2	4	20	0	0	0
52001	70	0	10	11	0	0	0
52100	70	1	21	22	0	0	0
52101	70	0	26	27	0	0	0
52102	70	0	29	30	0	0	0
150400	105	0	1	17	33	0	0
150700	106	0	3	19	35	0	0
60200	71	0	9	0	0	0	0
60300	72	0	10	11	12	0	0
60400	73	0	1	17	0	0	0
60500	74	0	14	15	16	0	0
61000	75	0	2	18	34	0	0
61200	76	0	2	18	34	0	0
61600	77	0	30	31	32	0	0
61700	78	0	30	31	32	0	0
62200	79	0	5	21	37	0	0
62300	80	2	1	17	33	0	0
62400	81	1	5	21	37	0	0
62500	82	0	10	26	0	0	0
62600	83	0	1	17	0	0	0
62700	84	0	2	3	4	0	0
63100	85	0	4	20	36	0	0

```
  66000     96     1    14    15    16     0     0
  66100     97     0     9    10    11    25    26
  66200     98     0     4    20     0     0     0
  66200     99     0    25    26    27     0     0
  71900     86     0    21    22    23     0     0
  73500     87     0     4    20     0     0     0
  73600     87     0    34    35    36     0     0
 170800    107     4     3    19    35     0     0
 170801    108     0     6    22    38     0     0
 171100    109     1     1     2    17    18    33
 171300    110     0     5     6    21    22    37
 171500    111     0     5     6    21    22    37


DONAGHEY   TABLE  395      36  62117     25     20  R N $    50.00
```

Figure 1 (cont.)

RESULTS WITH HIGH SCHOOL DATA

This technique was then attempted with data from Woodlawn Junior High School. This school had 507 students (each student requesting an average of 9 courses), and 193 course section offerings. A feasible master schedule was generated and assignments were made. On the first cycle 198 students were fully scheduled (they were scheduled to every course they requested). There were 650 course requests that could not be filled. So while only 39% of the students were given a full schedule using the feasible master schedule, 86% of the course requests were satisfied. The majority of the students who did not receive a full schedule missed only 1 course which could not be scheduled. There were no alternate requests for the students. For example, a student requested 9 courses, no substitutions were allowed. If possible, it would be desirable to have alternatives. In other words if the student cannot schedule French, an alternative course may be Spanish. However, in our discussions with high school administrators this is not practical in most cases.

Figure 2 shows how student schedules are printed by the computer. Student Robert E. Morris (Student No. 34) was scheduled for 10 courses. The next line shows the courses he requested 7061, 7051 etc. The following line shows the courses the machine was able to schedule for the student. Notice the student requested course 7350 and the machine scheduled him to 7351. The first 3 digits (735) are the course number and the last digit is the section. In this case the student could not be scheduled to the "zero" section, but was able to be scheduled to the "one" section. At Woodlawn the students were grouped according to I.Q.'s, and the administration wanted to retain these groupings for the academic courses, such as English and Mathematics. Student Morris is in the "one" group so he was required to request the "one" sections of courses 700, 705, 710, 720 and 725. Courses 735, 740, 760, 765 are Music, Art, Industrial Arts and Physical Education where the groupings do not apply. The next five lines of output for Robert Morris show his weekly schedule. On Monday the first period he has course 705, the second period Monday course 740, etc. On Tuesday the first period he again has 705, the second period Tuesday course 760, etc. Each line represents his daily schedule for Monday through Friday, and each element in the line represents the periods from one to eight. The school administrators at Woodlawn felt this output could be distributed, without further refinements, to the students for this weekly schedule. Students Havrilla and Kostilnik on the sample output could only be scheduled to 9 courses even though they requested 10 courses. In both cases the computer could not schedule them to course 765.

Problems were encountered when the computer attempted to re-schedule the course causing the greatest number of conflicts. No available time periods could be found to re-schedule this course. At Woodlawn the the faculty is scheduled to teach from 30 to 37 hours out

Figure 2

```
MORRIS ROBERT E          34      10

   7001    7051    7101    7151    7201    7251    7350    7400    7600    765
   7001    7051    7101    7151    7201    7251    7351    7401    7601    7651

    7051     7401     7101     7151     7651     7251     7001     7201
    7051     7601     7101     7151        0        0     7001     7201
    7051     7401     7101     7151     7651     7251     7001     7201
    7051     7601     7101     7151        0        0     7001     7201
    7051     7401        0     7351     7351        0     7001        0


REDPATH DALE R           35      10

   7001    7051    7101    7151    7201    7251    7350    7400    7600    7650
   7001    7051    7101    7151    7201    7251    7351    7401    7601    7651

    7051     7401     7101     7151     7651     7251     7001     7201
    7051     7601     7101     7151        0        0     7001     7201
    7051     7401     7101     7151     7651     7251     7001     7201
    7051     7601     7101     7151        0        0     7001     7201
    7051     7401        0     7351     7351        0     7001        0


HAVRILLA MARY E          36       9

   7001    7051    7101    7151    7201    7251    7300    7450    7600    76
   7001    7051    7101    7151    7201    7251    7301    7450    7601       0

    7051        0     7101     7151     7450     7251     7001     7201
    7051     7601     7101     7151     7450        0     7001     7201
    7051        0     7101     7151        0     7251     7001     7201
    7051     7601     7101     7151     7450        0     7001     7201
    7051        0        0     7301     7301        0     7001        0


KOSTILNIK DEBOR          37       9

   7001    7051    7101    7151    7201    7251    7300    7450    7600    765
   7001    7051    7101    7151    7201    7251    7301    7450    7601       0

    7051        0     7101     7151     7450     7251     7001     7201
    7051     7601     7101     7151     7450        0     7001     7201
    7051        0     7101     7151        0     7251     7001     7201
    7051     7601     7101     7151     7450        0     7001     7201
    7051        0        0     7301     7301        0     7001        0
```

of a 40 hour week. The times to which the courses could be re-scheduled were therefore limited from a faculty standpoint. There were also a large number of special purpose rooms (typing rooms, industrial arts, home economics, etc.), that were scheduled almost the entire week. In our case a 9th grade music course, that must meet in the music room, caused the greatest number of conflicts. In the only 5 free time periods the instructor had available where this course could be re-scheduled, the music room was occupied with another course. Therefore this worst course could not possibly be re-scheduled with our program. It was found that a large number of courses could not be re-scheduled, and even when a course could be re-scheduled there was a considerable amount of computer search time for the available free periods. Using the techniques outlined in the previous paper for adjusting the master schedule, the number of conflicts was reduced only 4% with 15 cycles and 2 hours of 7070 time. The reason for this poor performance, of course, was the very restrictive schedule at Woodlawn. However, we felt this situation is typical at most high schools and began to investigate for ways to improve our program to handle these cases.

We first looked at our system for generating a feasible master schedule. This original program produced more than satisfactory results for problems that were not too restrictive. In order to improve the program the next step was to take the student course requests into consideration in addition to the faculty and room requirements. Therefore, another program for generating a feasible master schedule was prepared for more restrictive cases.

A REVISED MASTER SCHEDULE GENERATOR

The names of the courses or course sections that should not conflict with each other are a part of the input to this program. This data can be generated manually or by another program from student requests. Theoretically we could feed in the course conflict matrix as the course conflict restrictions. However, this will increase the computer time tremendously since it is directly dependent upon the number of restrictions. Since each section of a course is treated as a unique course the course conflict matrix will be large. In the case of Woodlawn, the matrix would be of order 193 x 193. We can reduce the number of restrictions by putting the students on blocks and assigning these blocks to specific sections. In this case we would only have to write restrictions for each block instead of each student. In most cases it will not be possible to put all the students on these blocks; however, in high schools the large majority of the students can be blocked. The quality of the results of this program is greatly dependent upon the way that the restrictions are written. The job of putting the students on blocks and assigning them to sections is considered in another section of the final report.

The procedure followed in the program is not difficult. Different time patterns are built for different types of courses. One hour per week courses, two hours per week courses, laboratories, etc. First the type of the course or course section that we are trying to schedule is identified. Then all the time patterns available for that type of course are tried in sequence until a time pattern without any block, faculty, and room conflicts is found, and the course is assigned to that time pattern. If the computer tries all appropriate time patterns and finds none without conflicts a message describing the conflict is printed out and the computer goes to the next course.

The experiments that we had with Woodlawn High School indicated that we have to make our time patterns more flexible. For example we built our time patterns for three hour per week courses on Monday, Wednesday and Friday at the same time each day. Time patterns built on any three days at different hours will give the program more flexibility and produce more workable results. These time patterns will have to be within the limitations of the administrative policy of the school. When this program was tested with the Woodlawn data it was found that not a sufficient number of time patterns had been made available to the computer. As a result there were a large number of courses that could not be scheduled. We are now in the process of putting more time patterns in the program and are working on a technique where the school administrator can decide which time patterns will be acceptable to him.

AN IMPROVED ASSIGNMENT TECHNIQUE

Another way of improving our original program, would be to re-write the algorithm for assigning students, once the master schedule has been generated. The original assignment technique follows:

Assume the schedule for 3 courses, English, Spanish, and Algebra is

English	1st section	Monday	through	Friday	1st	period
"	2nd section	"	"	"	2nd	"
"	3rd section	"	"	"	5th	"
Spanish	1st section	"	"	"	1st	"
"	2nd section	"	"	"	2nd	"
"	3rd section	"	"	"	3rd	"
Algebra	1st section	"	"	"	1st	"
"	2nd section	"	"	"	2nd	"

Assume a student requests English, Spanish and Algebra, in this order, with no section preferences. Since English is his first request he would be assigned first to English and put in section one of that course provided the maximum number of students for that section has not been reached yet. If section one were filled the program would assign him to section two. Assuming that section one is not filled, then this student's weekly schedule at this point would be

PERIOD

	1	2	3	4	5	6	7	8
Mon.	Eng.Sec. 1							
Tues.	"							
Wed.	"							
Thurs.	"							
Fri.	"							

His second request is Spanish and the program would attempt to assign him to Section one of Spanish. This cannot be done because Spanish Section one conflicts with English Section one previously assigned. The computer would then consider Spanish Section two and if there is room in this section, the assignment would be made. His schedule now would be as below:

PERIOD

	1	2	3	4	5	6	7	8
Mon.	Eng. Sec. 1	Span.Sec. 2						
Tues.	"	"						
Wed.	"	"						
Thurs.	"	"						
Fri.	"	"						

Algebra, request three, is now considered. Algebra Section one cannot be assigned because of English, and Algebra Section two, the only remaining section, cannot be assigned because of Spanish. In the original assignment program a conflict would be tallied against Spanish, Section two and if this course caused more conflicts than any other course in the master schedule, it would be re-scheduled to a new time pattern. However, it is evident, that even with no adjustment to the master schedule, the student could still schedule the three courses. With the new program, given the same master schedule and student requests, assignment would proceed as follows:

The student would initially be assigned to English section one.

	PERIOD							
	1	2	3	4	5	6	7	8
Mon.	Eng.Sec.1							
Tues.	"							
Wed.	"							
Thurs.	"							
Fri.	"							

Next an attempt would be made to schedule him to Spanish, Section one, but this cannot be done since it conflicts with English Section one. However, before moving to Spanish, Section two the machine considers the previously scheduled course, in this case English, and looks for another available section of this course. It finds that English Section two can be scheduled, assigns the student to that section, and schedules him to the first section of Spanish. His schedule now appears:

PERIOD

	1	2	3	4	5	6	7	8
Mon.	Span.Sec. 1	Eng.Sec. 2						
Tues.	"	"						
Wed.	"	"						
Thurs.	"	"						
Fri.	"	"						

When Algebra is considered, the computer detects a conflict when it attempts to schedule the first section. An attempt is made to re-schedule the previously scheduled course, Spanish. Spanish Section two cannot be scheduled but the third section of Spanish can be scheduled the third period every day. This will then allow Algebra to be scheduled the first period. The students schedule, therefore is:

PERIOD

	1	2	3	4	5	6	7	8
Mon.	Alg. Sec. 1	Eng. Sec. 2	Span. Sec. 3					
Tues.	"	"	"					
Wed.	"	"	"					
Thurs.	"	"	"					
Fri.	"	"	"					

Assume the student had a fourth request, Music which has only one section scheduled for the first period Monday through Friday. Even with our improved program for assigning students, this student could not get the Music course. The computer would attempt to re-schedule Algebra, the course first scheduled to the first period, but the only remaining section of Algebra meets second period which is already blocked out by English.

To handle cases such as this the program was written to go back a second level of unscheduling in an attempt to give a student a full schedule. To show what is meant by second level unscheduling, consider again our recent example where the student could not schedule Music because another section of Algebra could not be found. When using second level unscheduling the computer will look at the course that prevents Algebra section two from being assigned. In this case it is English. It looks now for another section of English that would fit the student schedule. English Section three is available and the student would be assigned to that section (provided there is room). This would free the second period for Algebra, which in turn would allow Music to be scheduled first period. The student's schedule would now be:

PERIOD

	1	2	3	4	5	6	7	8
Mon.	Music	Alg.Sec. 2	Span.Sec. 3		Eng.Sec. 3			
Tues.	"	"	"		"			
Wed.	"	"	"		"			
Thurs.	"	"	"		"			
Fri.	"	"	"		"			

The new assignment program was tested with the Woodlawn data. With the original assignment routine 198 out of the 507 students obtained full schedules. The computer then went back and used 1st level in scheduling. From this pass the number of fully scheduled students went to 225. The machine then went through a third pass using second level unscheduling and increased the number of fully scheduled to 267. The three passes took 2-1/2 hours of 7070 computer time. The number of fully scheduled students jumped from 39% with our first system to 53% using second level unscheduling. Out of the 4500 course requests there were 367 (8.15%) that were not filled using second level unscheduling.

FURTHER RESEARCH

The heuristic approach outlined in this report can be utilized as an effective solution to the problem of generating a master schedule and assigning students. As a result of our experience in the field testing, work in the following area is suggested.

1. Build more flexible time patterns in the master schedule generating program.
2. Devise a technique where school administrators could easily inform the computer which time patterns are acceptable, and perhaps attach preference weights.
3. Investigate the feasibility of going to third and fourth level unscheduling.
4. Rewrite the assignment program. The program, as it is now, is the result of two years work, and has constantly been revised, and additions made as new ideas were generated. As a result, the program is not efficient as far as computer time is concerned.

This can be attested, from the 2-1/2 hours it took on the final run with the Woodlawn data.

5. Adapt all the programs for the 7090. This computer has recently been installed at the University of Pittsburgh. It's speed is from 3 to 15 lines faster than the 7070; consequently, the cost of scheduling on the computer would be within the budget of most school districts.

5. CONCLUSIONS AND IMPLICATIONS

The following listing gives a brief overview of the Conclusions and Implications. For a more extended discussion refer to the section on Procedure, Analyses of Data, and Findings.

1. A feasible solution space which does not violate any of the constraints of the problem can be generated by Boolean Isomorphic Structures. It was shown in a 10 variable problem that out of 1024 possible configurations, only 102 of these configurations are feasible. As the size of the scheduling problem increases, the number of possible combinations for time assignments based on a selected criterion becomes excessively large.
2. The Relations Matrix, developed from the constraints imposed on the scheduling system, can be used effectively to determine the minimum number of time period vectors required to satisfy the constraints. This matrix is of dimension n x n, where n is the number of course sections. The system developed can accommodate 2000 course sections on a large size computer.
3. The model of an educational scheduling system can be decomposed so that an optimal master schedule cycle can be determined. This research decomposes the system into five phases, starting with the determination of the course sections to be offered, and terminating with a master schedule which satisfies the conditions established. The constraint structures imposed on the system are developed by the computer, presupposing

that an assignment of specific course sections to students, faculty, and physical plant facilities, has been made. The program has been documented, and at present can handle on the IBM 7090 up to 1000 course sections, with an unlimited number of constraints.

4. For the more realistic problem, where the schedule cycle length is controlled (fixed), the scheduling system developed to optimize the master schedule cycle can be relaxed so that the controlled cycle criterion is not violated. The criterion "no conflicts permitted for elements of the system" is relaxed to "minimum number of conflicts permitted." The criterion "master schedule must have a minimum number of days per cycle" is changed to "master schedule must not require more than a specified number of days in a schedule cycle." Again, this program has been documented, and can accommodate on the IBM 7090 up to 1000 course sections, with an unlimited number of constraints.
5. The controlled schedule cycle concept can be extended to construct an organized matrix of forced conflicts, similar to the Relations Matrix. The Forced Conflicts Matrix could then be used to separate the faculty, students, and physical plant facilities.
6. The constraints of an educational scheduling model can be cast as a linear programming model. This was actually accomplished for a senior high school.

7. The concept of a shadow price as used in linear programming can be extended to the school scheduling problem so that the administrator can evaluate marginal returns of potential changes in the system.
8. Scheduling problems with no more than 1024 constraints and any number of variables can be solved with the LP/90 program developed by CEIR. This program was made operational in the field testing of the linear programming approach. The system developed presently limited to schools with a master schedule cycle of one day, with minor exceptions permitted. While the limitation was discouraging at first, subsequent research, however, revealed that, many of the high schools in Allegheny County, Pennsylvania, are presently using a one day schedule cycle.
9. Existing computing equipment cannot accommodate the size of most real life scheduling problems for solution by linear programming. Techniques for surmounting or circumventing the size problem must be developed. A decomposition approach should be investigated.
10. The heuristic program, based on the development of a master schedule which contains no faculty or room conflicts but which reschedules the course causing the greatest number of conflicts, does perform satisfactorily when the problem is not highly restrictive. This was established from the results of testing in a university and a high school.

11. For the more restrictive cases, it was found that student course requests had to be considered as well as faculty and room requirements in constructing the original master schedule.

12. With time patterns built into the program, the experimentation conducted pointed up the need for flexibility in these time structures.

13. The number of fully scheduled students increased significantly as a result of an assignment technique which unscheduled two levels. Out of 4500 course requests in the high school field test, 91.85% of these requests were filled using second level unscheduling.

14. The underlying theory developed for a Probabilistic Aggregate Approach should prove helpful in further research on the scheduling problem. The overall intent is to provide an assignment procedure that will be computationally feasible, and adaptable to many educational configurations and requirements. If a given number of iterations is allowed with a stated schedule density, then it is desirable to be able to make a probability statement as to the reduction in conflicts for this given number of iterations. In economic terms, for X computer dollars, a specific conflict can be eliminated in the scheduling system.

15. The assignment of students to specific course sections in multi-section courses is critical in the construction

of master schedules. The generated master schedule is contingent upon the specific course sections assigned to the students. Programs have been developed for various assignment criteria, such as balancing the sections, homogeneous and heterogeneous groupings, and combinations of these criteria. It is ironic that the scheduler encounters the most serious challenge when he is given complete freedom in the assignment of multi-section courses to students.

16. In the total systems concept of the scheduling function, feedback between the scheduling system and the student's master record is essential to obtain full schedules for all students. For unresolved conflicts, it is desirable to have the computer generate new course requests using the same criteria for selection as that of the adviser and student. In this research an operational system has been developed to generate course requests and provide file maintenance on a master tape. A closed loop system should be developed between the scheduling system and the master record file for real time feedback.

6. APPENDIX

PROBABILISTIC AGGREGATE APPROACH

BY

P.J. KIVIAT

INTRODUCTION

The task of any student scheduling procedure must be to produce efficient or "optimal" sets of schedules in an efficient and economic manner. Although it is theoretically possible to construct optimal scheduling procedures (relative to some well-defined measure), the combinatorial aspects of most practical problems make such procedures computationally infeasible. It is necessary to investigate procedures for generating schedules that somehow reduce the dimensionality of the scheduling problem. This report describes one such procedure.

In his thesis, Computer Learning Theory in University Scheduling, Brubaker has shown that it is not possible for a scheduling procedure to consider all the possible schedules that might arise from a given student scheduling problem and arrive at optimal answer in a reasonable amount of time. The approach of the procedure described here is one of aggregation of student course demand, and creation of a Master Schedule to which individual students are assigned. The Master Schedule is a probabilistic reflection of individual student demand; since it is derived from pre-registration data it is subject to error due to last minute student changes, but unless these are large in number, and drastic in deviation from previous course requests this error will most likely be small. The procedure generates schedules based on a-priori estimates of student demand; it then assigns individual students by means of an iterative assignment algorithm which converges to a best overall student assignment.

GENERATION OF A "MOST-LIKELY" MASTER SCHEDULE

Let $C = (c_1, c_2, \ldots, c_n)$ be a set of courses offered by an educational institution, and $S = (s_1, s_2, \ldots, s_m)$ be a set of students at this institution requesting courses for the coming scheduling period.

Define a schedule group G, as a pattern of courses of a similar nature. $G = (g_1, g_2, \ldots, g_k)$ and is a partition on C.

Define a student schedule group $\mathscr{S}$, as a set of Boolean vectors indicating individual student schedule group requests. $\mathscr{S} = (\mathscr{S}_1, \mathscr{S}_2, \ldots, \mathscr{S}_m)$; $\mathscr{S}_i = (\mathscr{s}_{1i}, \mathscr{s}_{2i}, \ldots, \mathscr{s}_{ki})$ where $\mathscr{s}_{ji} = 1$ if student i requests a course in schedule group j, and $\mathscr{s}_{ji} = 0$ otherwise. S is a matrix of order m x k.

Define a student schedule group conflict pattern P as the matrix generated by the matrix product of S^T and S, i.e., $P = S^T \cdot S$. An element of P, p_{ij}, is the schedule group conflict of the ith schedule group with the jth schedule group. $p_{ij} = \sum_{p=1}^{m} (S_{ip})^T S_{ip}$; for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$. The conflict relation is non-reflexive, symmetric and transitive; however, the normal Boolean product is retained in the generation of the matrix.

Interpret P as follows:

> tr(P) = total number of schedule group requests. Since it is possible for a student to select more than one course in a schedule group tr(P) = total number of course requests.

> $P_i' = \frac{p_{ii}}{tr(P)}$ = Prob (student will request at least one course in schedule group i).

Since this probability is developed from a pre-registration list, and therefore subject to error due to request changes it is deemed an a-priori probability. If there are relatively few request changes and/or additions after the preregistration period this a-priori probability will still reflect the aggregate student demand. The development of the "most-likely" Master Schedule is dependent upon this concept of a probability of incurring certain student request patterns.

$$p'_{ij} = \frac{p_{ij}}{\sum\limits_{j:j\neq i} p_{ij}}$$ = Prob (student will request at least one course in schedule group $\underline{j}$ given that he has requested a course in schedule group $\underline{i}$).

= a-priori conflict probability between schedule groups $\underline{i}$ and $\underline{j}$.

To generate the matrix P from student course requests:

(1) Each student submits a request for certain courses for the coming scheduling period.

(2) According to the partition G of C, the student schedule group vectors are generated from the student course requests. It must be noted that this partition is of crucial importance to the success of this procedure; more will be said of it later.

(3) The student schedule group conflict pattern P is generated from the student schedule group vectors.

The partition G has allowed us to reduce the number of different course (student) schedules from 2^n-1 to 2^k-1, which in turn has allowed us to compute P. We now use P to assign the schedule groups to time periods to form the Master Schedule.

Consider the schedule time as consisting of "horizontal" and "vertical" time patterns; a time period refers to distinct elements of either pattern. If there are fewer schedule groups than time periods the assignment is trivial; if there are more there are three possible schedule group interactions: separation, overlap, or covering. The ratios n/k and k/t influence the interaction pattern that will be present in the developed schedule; t is the number of time periods in the time pattern under consideration.

Consider the non-trivial case. Assume that any schedule group can be assigned to any time period. To develop the Master Schedule:

(1) Assign the schedule group with the highest a-priori probability of being requested to the first time period, call it p_j'. Then $p_j' = \max (p_i')$; $i = 1, 2, \ldots, k$.

(2) Assign the schedule group with the next highest a-priori probability of being requested to the next available time period, call it p_q'. Then $p_q' = \max (p_i')$; $i = (1, 2, \ldots, j - 1, j + 1, \ldots, k)$. Continue assigning schedule groups until all time periods are filled. There are now k - t schedule groups remaining to be assigned to time periods. This assignment has the desirable feature of tending to level facilities and staff requirements during maximum request times.

(3) Find the schedule group of those not yet assigned that has the highest a-priori probability of being requested. Compute the conditional probabilities p_{ij}', of each of the previously assigned schedule groups being requested, given that this schedule group is requested. Assign

this schedule group to the same time period as the schedule group with the minimum a-priori conflict probability. This assignment tends to reduce the probability of course conflicts during final student assignment to the Master Schedule.

(4) Continue until all schedule groups have been assigned to time periods in this time pattern.

(5) Repeat this procedure for all time patterns, within each higher level time pattern assignment.

It can be noted here that a sequential assignment rule is used; at each level (time pattern) a time period assignment is made, each succeeding level of assignment is "best" relative to its preceding assignment.

An example will be used to clarify this procedure:

$C = (c_1, c_2, \ldots, c_{50})$ 50 courses

$G = (g_1, g_2, \ldots, g_5)$ 5 schedule groups

$t_1 = 2$, "horizontal" pattern; MWF, TTS

$t_2 = 2$, "vertical" pattern; AM, PM

student schedule group vectors; generated from student course requests.

$$\mathcal{S} = \begin{array}{lccccc}
s_1 = & 1 & 1 & 0 & 1 & 0 \\
s_2 = & 1 & 1 & 0 & 0 & 1 \\
s_3 = & 0 & 1 & 1 & 1 & 1 \\
s_4 = & 1 & 1 & 1 & 1 & 0 \\
s_5 = & 0 & 1 & 1 & 1 & 1 \\
s_6 = & 1 & 1 & 1 & 0 & 0 \\
s_7 = & 0 & 0 & 1 & 1 & 1 \\
s_8 = & 0 & 1 & 1 & 1 & 0 \\
s_9 = & 1 & 1 & 1 & 0 & 0 \\
s_{10} = & 1 & 0 & 0 & 1 & 0
\end{array}$$

$$P = \begin{bmatrix} 6 & 5 & 3 & 3 & 1 \\ 5 & 8 & 6 & 5 & 3 \\ 3 & 6 & 7 & 5 & 3 \\ 3 & 4 & 5 & 7 & 3 \\ 1 & 3 & 3 & 3 & 4 \end{bmatrix} \quad ; \ tr(P) = 32$$

The request probabilities:

$$\begin{array}{ll} p'_1 = 6/32 & p'_4 = 7/32 \\ p'_2 = 8/32 & p'_5 = 4/32 \\ p'_3 = 7/32 & \end{array}$$

The conflict probabilities:

$$\begin{array}{llll} p'_{12} = 5/12 & p'_{13} = 3/12 & p'_{14} = 3/12 & p'_{15} = 1/12 \\ p'_{21} = 5/19 & p'_{23} = 6/19 & p'_{24} = 5/19 & p'_{25} = 3/19 \\ p'_{31} = 3/17 & p'_{32} = 6/17 & p'_{34} = 5/17 & p'_{35} = 3/17 \\ p'_{41} = 3/15 & p'_{42} = 4/15 & p'_{43} = 5/15 & p'_{45} = 3/15 \\ p'_{51} = 1/10 & p'_{52} = 3/10 & p'_{53} = 3/10 & p'_{54} = 4/10 \end{array}$$

The first assignment is to the horizontal time pattern MWF, TTS:

(1) max $(p'_i) = p'_2 = 8/32$; assign g_2 to MWF, and set $p'_2 = 0$.

(2) max $(p'_i) = p'_3 = p'_4 = 7/32$; assign g_3 to TTS, since $p'_{14} = p'_{13}$, and $p'_{41} > p'_{31}$. Set $p'_3 = 0$.

(3) All time periods in the current time pattern are now filled; use the conflict probabilities (as in (2) above) to make the remaining assignment decisions.

max $(p'_i) = p'_4 = 7/32$; min $(p'_{4j}) = p'_{42}$, of those g_i that have already been assigned to time periods. Assign g_4 to MWF which is the time period that g_2 has been assigned to, and set $p'_4 = 0$.

(4) max $(p_i') = p_i' = 6/32$; min $(p_{1j}') = p_{13}' = p_{14}'$, of those g_i that have already been assigned to time periods. Assign g_1 to TTS which is the time period that g_3 has been assigned to, and set $p_1' = 0$; attempt to minimize number of groups per time period.

(5) g_5 remains unassigned. Min $(p_{5j}') = p_{51}'$, assign g_5 to TTS which is the time period that g_1 has been assigned to, and set $p_5' = 0$.

Now, within the horizontal time pattern assignment, assign the schedule groups to the vertical time pattern AM, PM:

(6) Consider the time pattern AM, PM within the pattern MWF: g_2 and g_4 have been assigned to MWF; following steps (1) thru (5) above we arrive at the assignment of g_2 to MWF, AM and g_4 to MWF, PM.

(7) Consider the time pattern within the pattern TTS: g_1, g_3 and g_5 have been assigned to TTS; as in (6) above we arrive at the following assignment: g_3 to TTS, AM, g_1 to TTS, PM and g_5 to TTS, PM, since $P_{51}' < P_{53}'$.

The basic scheduling unit in a University class schedule is the course section, at this point no mention has been made of this unit; by concentrating our attention on a much coarser division, the schedule group, we have been able to arrive at a very rough Master Schedule. We have seen that in arriving at this Schedule we have made several simplifying assumptions: 1) homogeneity of a schedule group-time period-time pattern associations, 2) equality of schedule groups, i.e., no schedule groups are "more important" than others, although this would be a simple addition to the assignment rules, and 3) most important of all, the existence of a feasible partition G of C, that is, dependence on a "natural" grouping of courses within schedule groups.

To refine the Master Schedule to the course section level we must rely on the ability of assignment algorithm to be used recursively. It was stated that the primary reason for the schedule group partition was to reduce the dimensionality of the scheduling problem; this allowed us to compute the a-priori request and conflict probabilities, and to develop a Master Schedule based on these probabilities. A heuristic dimensionality reduction principle allowed us to develop the Master Schedule in a sequential assignment through time pattern levels. We now rely upon this principle to further refine the Master Schedule.

Conflicts can arise in the current Master Schedule only when two or more schedule groups occupy the same, lowest level, time period. We now consider further division of the lowest time pattern considered so far, and instead of considering all schedule groups, we consider only those within the lowest level time pattern generated so far. Since there will be far fewer schedule groups in this new problem we may now expand our original partition G to a more workable level. We now assign our new scheduling units within their respective time patterns.

In the above example, the Master Schedule that was created assigned schedule groups, first to weekday time patterns, and then to morning and afternoon time periods; there was one schedule group conflict for one student in the resulting student schedules. However this says nothing of possible conflicts of courses within schedule groups; if no student ever takes more than one course in any schedule group, then all courses within each schedule group may be assigned the same schedule time, however it is too much to hope that this will be true. If we divide both AM and PM time patterns into four hourly divisions, then we can assign courses within schedule groups to minimize these conflicts. As mentioned previously, we have the option at this point of overlapping course sections if we choose.

The coarseness of the repartition G' of C must necessarily be a function of the partition G of C, and the number of schedule groups within the time periods being reassigned. We again strive for a partition that does not impose unduly high dimensionality demands on the algorithm.

For the above example:

$$\frac{\dim(C)}{\dim(G)} = \frac{50}{5} = 10; \; t_1 = t_2 = 2;$$

We wish to consider the new time pattern $t_3 = 4$; to retain the dimensionality of the first problem we must repartition C such that $[\dim(g_i)/\dim(g_i')]\,[2/4] \leq 10$ for all g_i within any time period being rescheduled. We may define the quantity $\frac{\dim(C)}{\dim(G)} t_j$ as the schedule density for the jth time pattern-course partition assignment pair.

It is now necessary to consider each lowest level time period, AM or PM for example, as a separate scheduling problem. The student course requests must be reconsidered in view of the new partition G', and new request and conflict probabilities developed. The assignment algorithm is then applied to each problem separately, and repeatedly until a final Master Schedule is developed.

The scheduling procedure then consists of 1) an assignment algorithm that utilizes request and conflict probability matrices, 2) a sequence of partitions, G, G', G'', ... G''''' of C, that generate the matrices, and 3) a schedule density that determines the partition sequence.

DIRECTION OF FURTHER RESEARCH

(1) Construction of a computer program for use in investigating properties of the assignment procedure.

(2) Investigation of the effect of the schedule density on the time required for schedule generation. It is expected that as the schedule density increases the generation time will increase, but in an as yet unknown manner.

(3) Study of what have been called "natural" partitions of C, and the development of computer learning or heuristic routines for generating these partitions.

(4) Development of efficient computer routines for implementing the procedure.

(5) Investigation of the convergence properties of the assignment algorithm. If we specify the minimum number of conflicts in all student schedules as our criteria of schedule performance, then there is some minimal conflict level which is theoretically attainable. Application of

the assignment algorithm at different schedule densities and partitions will produce schedules with different conflict patterns; we wish to find the optimal searching technique to arrive at the best schedule that the algorithm can generate. At that point the number of conflicts resulting from the program generated schedule will most likely not be optimal, since the schedule was generated from the aggregate characteristics of data that was subject to change. Further decreases in conflicts may then be obtained by proper schedule revision; which leads to:

(6) Development of a heuristic procedure, which when applied at any level of Master Schedule generation will tend to smooth out the conflicts resulting from the aggregation of the input data.

The overall intent is to provide an assignment procedure that will be computationally feasible, and adaptable to many educational configurations and requirements. The procedure will have the property of converging on a minimal conflict number as it is applied in successive steps. Research on the convergence properties will allow optimal convergence rates. Thus, if a given number of iterations is allowed with a stated schedule density we should be able to make a probability statement as to the reduction in conflicts for this given number of iterations. In economic terms, for X computer dollars, a certain conflict can be eliminated for an institutional scheduling problem.

OTHER RESEARCH

A. A recent paper by Eugene S. Schwartz of the Armour Research Foundation, "An Automatic Sequencing Procedure with Applications to Parallel Programming," *Journal of the Association for Computing Machinery*, October, 1961, comes very close to satisfying the requirements of an educational institution scheduling procedure. It is an automatic sequencing procedure based on a graph-theoretic model of a flow chart; it is concerned with satisfying priority requirements inherent in computer programs, and of avoiding interaction, and minimizing queueing problems which result from several programs being run on one computer simultaneously. The procedure consists of the construction of a *precedence matrix* from computer program (student schedule) requirements, and the transformation of this matrix into an *assignment matrix*. A set of assignment rules then establishes module (student) sequences and facility (class, section, faculty) loadings; the assignment rules do not guarantee an optimal assignment, but do guarantee a feasible one. The procedure is one of the local optimization of logically distinct parts of a problem by the application of certain assignment rules; while each rule results in an optimum solution for any particular application, the overall combination of results is not necessarily optimum. Trial applications of the procedure have resulted in optimum or near optimum results.

While conceptually applicable to the student scheduling problem, the procedure appears to suffer from the "dimensionality problem." The procedure considers individual units in its assignment; this would require a matrix of dimension equal to [dim (S) dim (C)]^2 in the notation of the previous section.

This has the obvious consequence of being prohibitive in computer time and storage requirements. However, the procedure has much to contribute to scheduling theory in its approach, and it is quite possible that it may be applicable in the solution of particular parts of the overall scheduling problem; for example, in the latest stages of the previous procedure.

B. In an as yet unpublished paper, "Choosing an Optimal Subset of Product Sizes for Inventory: A Dynamic Programming Approach," C. R. Frank presents a dynamic programming approach to the selection (or conversely the elimination) of a subset of items from a product mix, for inventory stockage. From a given set of n + 1 items the dynamic programming approach selects the best 1, 2, ..., n + 1 -q items to stock, in that order, and evaluates the cost of not stocking the items at each stage.

In the student scheduling problem the problem of eliminating students, faculty, and/or courses to resolve conflict situations often arises. If the decision to eliminate a particular student, say, is made a purely local decision, then a clearly non-optimal schedule results. Heuristic procedures have been designed that collect information for use in more global decision procedures, and these tend towards being better decision makers than the strictly local procedures. It is suggested that a completely global approach of the nature of Frank's inventory selection procedure might be used in deciding what courses, students, and/or faculty are to be removed from scheduling problems, and in what order they should be removed for a most efficient schedule generation.

ASSIGNMENT OF STUDENTS
TO
COURSE SECTIONS
IN
AN EDUCATIONAL SCHEDULING PROBLEM

OSMAN S. OZKUL

INTRODUCTION

In a typical high school or university master schedule courses have multiple sections. It is often desired by the school administration to assign students to specific sections before time periods are assigned to courses. These assignments have been made on the basis of various criteria.

ASSIGNMENT CRITERIA

1. Assignment of students to course sections in accordance with their faculty preference. (Student-teacher ties) In this case faculty assignments must be made before the student assignments.
2. Balancing the sections with respect to the number of students. Here each section will have approximately the same number of students.
3. Homogeneous grouping of the students within the sections. The school administration may desire to have approximately the same level of achievement for members of a given section and thus assign students to sections according to a test score, such as IQ or quality point average.
4. Heterogeneous balancing of the sections with respect to the different groups in the student body. These groups may be liberal arts oriented students, science oriented students, and professional oriented students; or the groupings may simply be boys and girls. Mixing the students with varying backgrounds in the sections may stimulate interaction among the different groups

and prove to be more fruitful for discussion types of courses.

5. Combination of homogeneous grouping with the criterion of balancing. (3 and 2)
6. Combination of homogeneous grouping with heterogeneous balancing. (3 and 4)

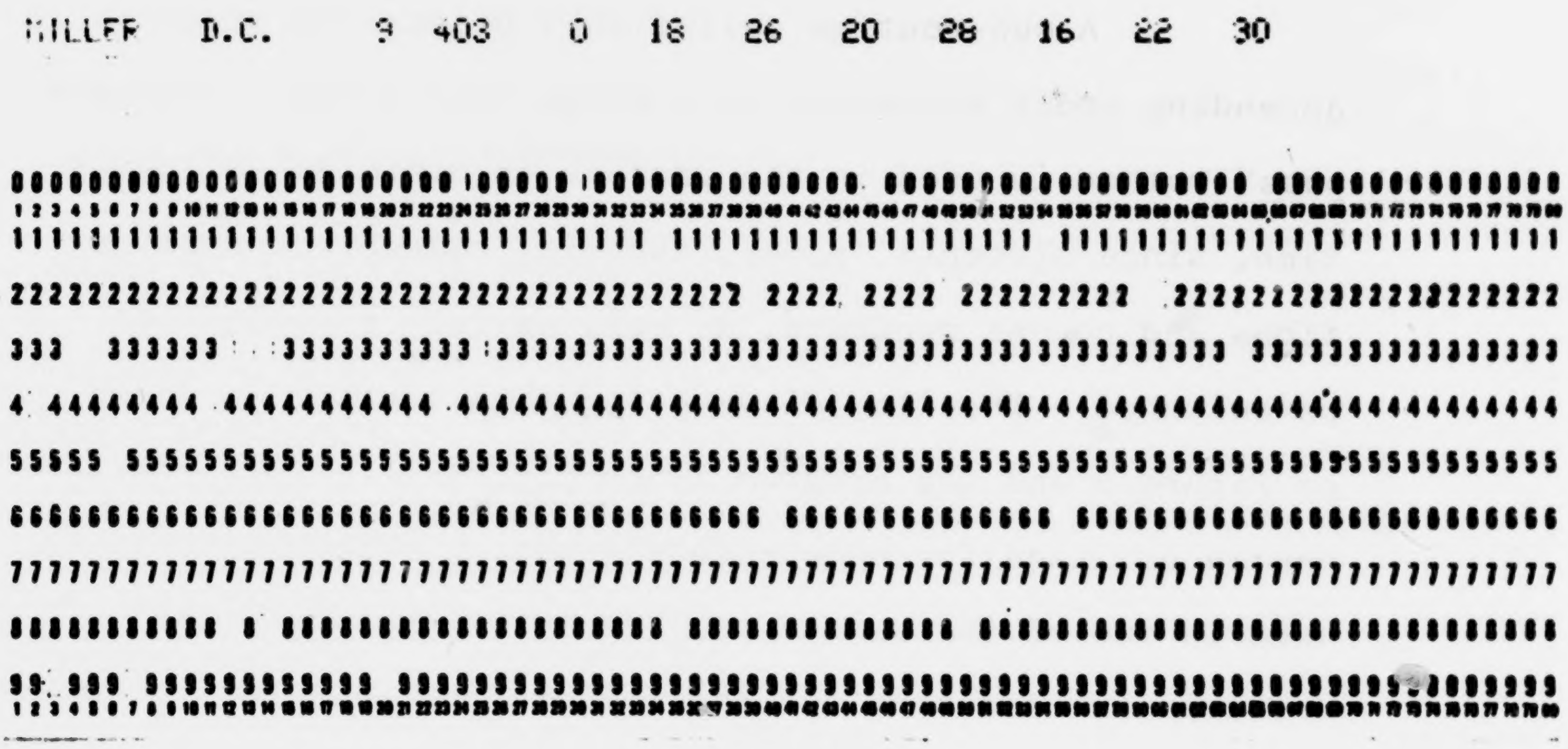
All of the above criteria are the results of some administrative policy. As far as the scheduler is concerned a criterion that helps to arrive at a workable solution of the scheduling problem may be the most desirable one to use.

7. Scheduler's criterion. Minimize the number of constraints generated by the student assignments to sections.

COMPUTER PROGRAMS FOR ASSIGNMENT CRITERIA

Computer programs have been written in this research project to assign students to specific sections according to criteria (2), (3), (4), (5) and (6). No program is necessary for the (1)st criterion since a student will ask for a specific course section on his course request card. A program will be written in the near future for the (7)th criterion.

The routines for criteria (2) through (6) have been put into a program package so that any desired criterion may be used. The criterion to be used is specified on the first input data card. The main part of the input data is the student request cards. Appropriate information for the criterion selected should appear on these cards. Here is an example of a typical student request card.



MILLER D.C. 9 403 0 18 26 20 28 16 22 30

Columns

1-15	Students' name
16-20	Students' number
21-25	Test score
26-30	Category specification
31-35	First course request
36-40	Second course request
•	•
•	•
•	•
76-80	Tenth course request

Ten hypothetical student request cards are made (Figure 15) and these sample students are assigned to sections. The actual computer outputs for four of the above seven criteria are shown in Figures 16 through 19. References will be made to these outputs later in this report.

A sub-routine called SORT places the students in ascending order according to a given test score. "Minimum pass" method is used in the process to minimize switching time, since students' names, numbers, category specifications and course requests, as well as test scores, must be switched. The flow diagram for this routine is shown on Figure 5 and the program is on Figure 12. This routine can be utilized for homogeneous grouping criterion. It is used in two of the sample problem solutions which use criteria (5) and (6), and the computer outputs for these are given in Figures 18 and 19. If the student requests are on cards, ordering can be accomplished manually on sorter. Depending on the number of students, it may be more economical and practical to do so. In this case the computer can skip this routine.

Three routines tally the number of requests for each course. Sub-routine ONE is used for non-grouped student bodies. It gives the total number of requests for each course. The flow diagram and the program for this routine are shown on Figures 1 and 6 respectively. This routine is used for solving the sample problem according to the criterion (2) and (5). The computer outputs for these solutions are Figures 16 and 18. Sub-routines TWO and THREE can be used for student bodies that are divided into two or three groups respectively for the purpose of heterogeneous balancing criterion (4). By using these routines we obtain the number of requests for each course from each of the groups of students. The flow diagram for sub-routine TWO is on Figure 2 and the programs for sub-routines TWO and THREE are on Figures 7 and 8. Because the students were put into one of the two categories in the sample problem, the use of sub-routine TWO is illustrated in solutions for criteria (4) and (6). The computer outputs for these appear on Figures 17 and 19. Sub-routine THREE would have been used in the same way for three categories of students.

At this point it is advisable to make the decisions regarding to the number of sections to be offered for each course manually. A program to accomplish this task on the computer can be written, but it may not be practical. After deciding on the number of sections to be offered for each course, any one of the next set of three routines can be utilized for actual assignment of the students to specific sections depending on the criterion that is used. The number of sections to be offered for each course for the sample problem is assumed.

Sub-routine ONEP assigns approximately the same number of students into each section and thus can be used for criteria (2) and (5). This is illustrated for sample problem on Figures 16 and 18. The flow diagram and the program for this routine are shown on Figures 3 and 9. Sub-routines TWOP and THREP are used for heteregeneous balancing criterion, i.e. balancing the different groups of students within the sections. If there are two categories of students sub-routine TWOP is used, and if there are three sub-routine THREP is used. Suppose there are two groups of students - the liberal arts oriented and the science oriented students. If 60 from the first and 40 from the second group of students requested a given course, 60/40 ratio will be maintained in forming the sections of this course by sub-routine TWOP. For the sample problem sub-routine TWOP is used for criteria (4) and (6). The computer outputs are on Figures 17 and 19. Sub-routine THREP does the assignment using the same techniques for three categories of students. The flow diagram for sub-routine TWOP is on Figure 4. The programs for sub-routines TWOP and THREP are on Figures 10 and 11.

Whenever a combination of two different criteria is taken as the desired criterion, in case of conflict, one of the criteria will take precedence over the other. For the (5)th criterion, combination of homogeneous grouping with balancing, in case of conflict, the criterion of balancing will take precedence over homogeneous grouping. For the (6)th criterion, combination of homogeneous grouping with heterogeneous balancing, in case of conflict, the criterion of heterogeneous balancing will take precedence over homogeneous grouping.

FUTURE RESEARCH

As mentioned earlier no computer program has been written as yet for the (7)th criterion. The following investigation is expected to be conducted relative to this criterion.

A Relations Matrix* will be generated while assigning students to sections. A zero at the intersection of two course sections in this matrix means that these two course sections should not be offered at the same time period. In order to overcome the over-constrained nature of the scheduling problem, the object of this program will be to minimize the number of zeros that goes into the relations matrix during the assignment phase of students to sections. The room, faculty and policy constraints that are known will be put into this matrix before the student assignments begin. When a student is being assigned to a course all the sections will be checked and the student will be assigned to an available section that generates the least number of additional constraints in the relations matrix. It seems that the criterion of balancing can be incorporated into this system very effectively whenever there are ties among the sections as far as the number of newly generated constraints is concerned. It is conceivable that the criterion of homogeneous sectioning also can be combined with the system to some extent.

---

*Relations Matrix Structures by I.B. Turksen and A.G. Holzman.

FLOW CHARTS

Input and output statements are omitted in the following flow diagrams. In order to conserve space the flow diagrams of sub-routines THREE and THREP are also omitted; in effect they are similar to sub-routines TWO and TWOP respectively.

SUB-ROUTINE ONE (TALLY NO. 1)

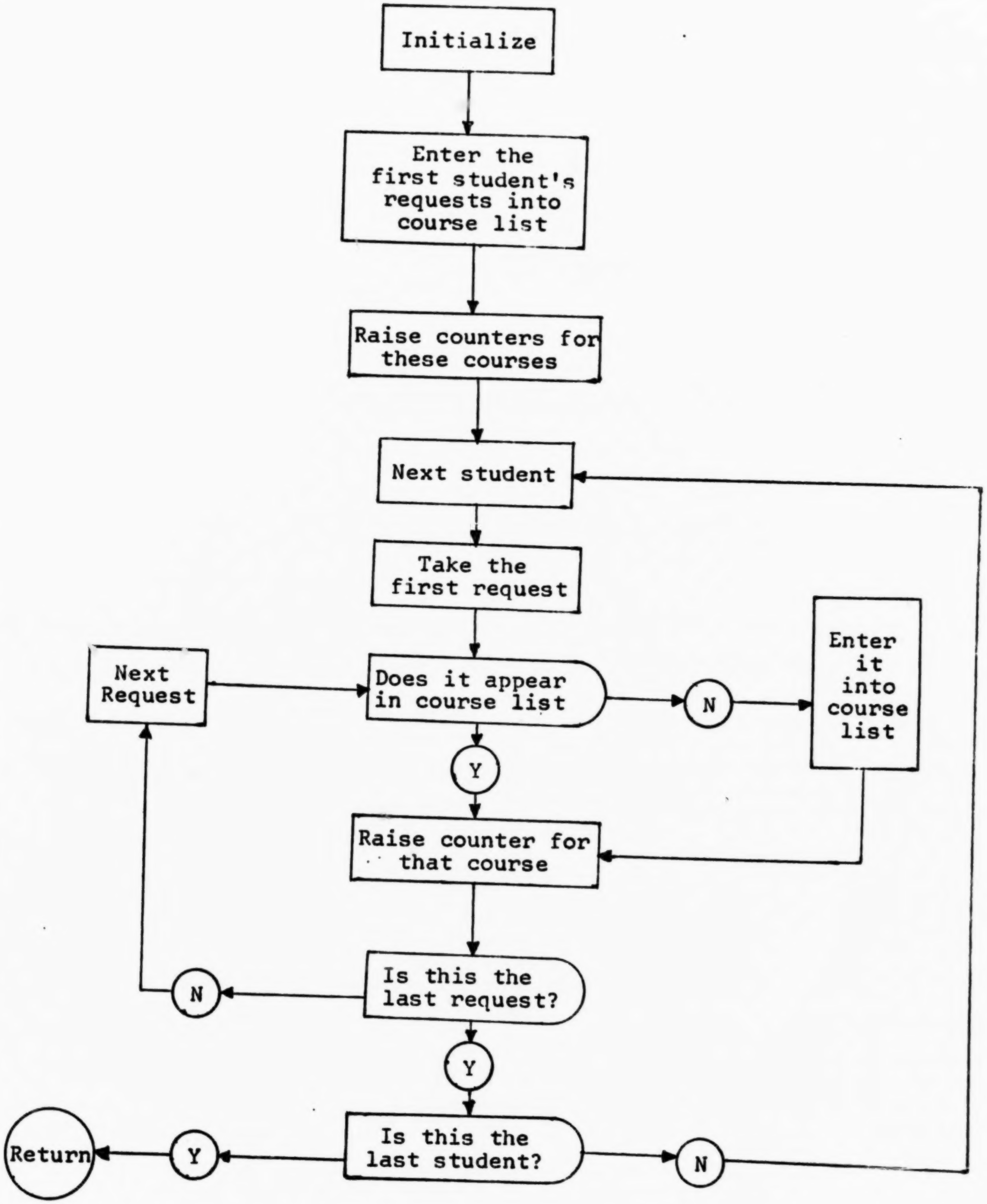

Initialize
Enter the first student's requests into course list
Raise counters for these courses
Next student
Take the first request
Next Request
Does it appear in course list
N
Enter it into course list
Y
Raise counter for that course
Is this the last request?
N
Y
Return
Y
Is this the last student?
N

Figure 1

SUB-ROUTINE TWO (TALLY NO. 2)

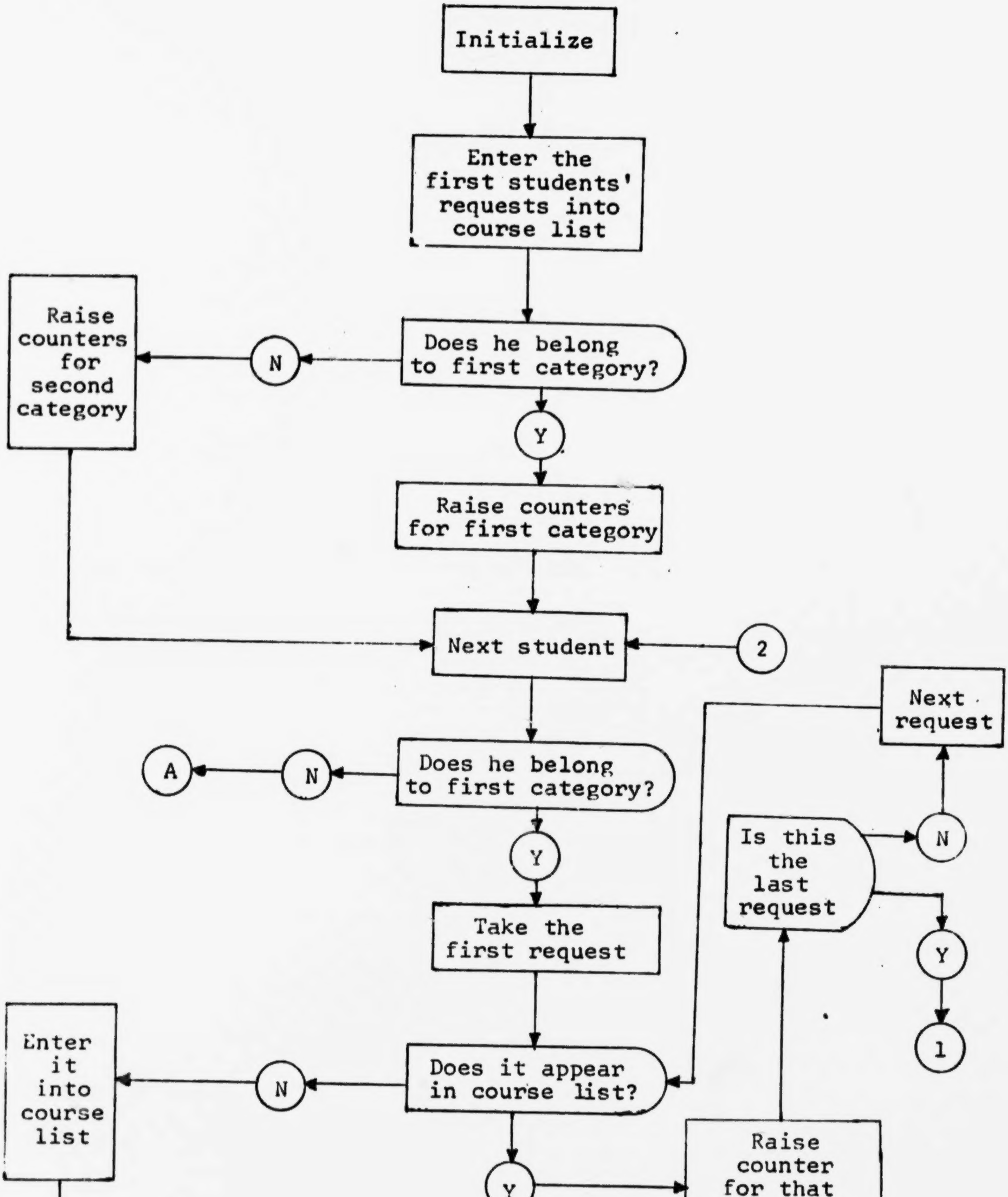

Initialize
Enter the first students' requests into course list
Does he belong to first category?
N
Raise counters for second category
Y
Raise counters for first category
Next student
2
Next request
Does he belong to first category?
N
A
Y
Is this the last request
N
Y
1
Take the first request
Enter it into course list
N
Does it appear in course list?
Y
Raise counter for that course for 1st category

Figure 2 - A

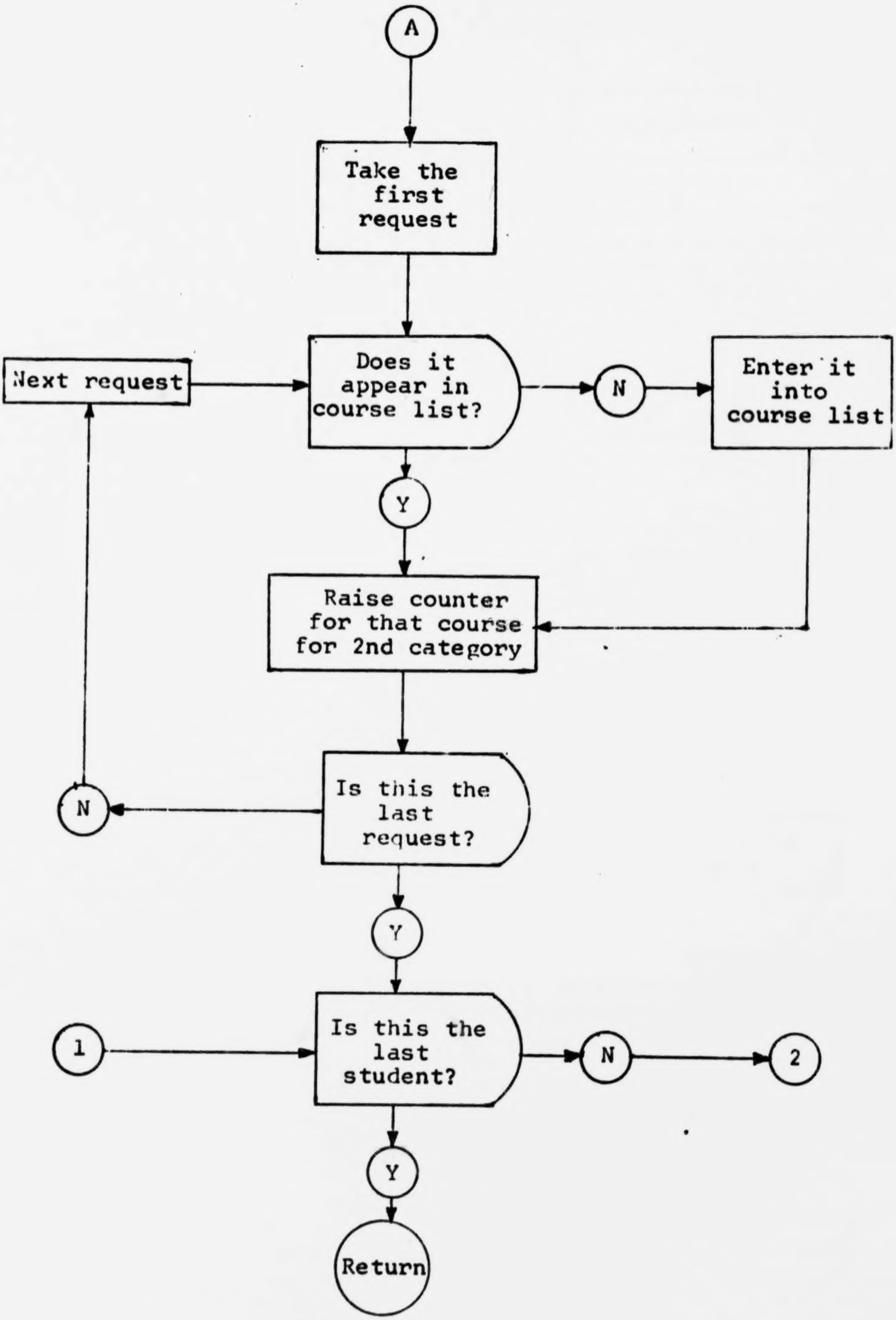

A
Take the first request
Does it appear in course list?
N
Enter it into course list
Next request
Y
Raise counter for that course for 2nd category
Is this the last request?
N
Y
1
Is this the last student?
N
2
Y
Return

Figure 2 - B

SUB-ROUTINE ONEP (ASSIGNMENT ROUTINE NO. 1)

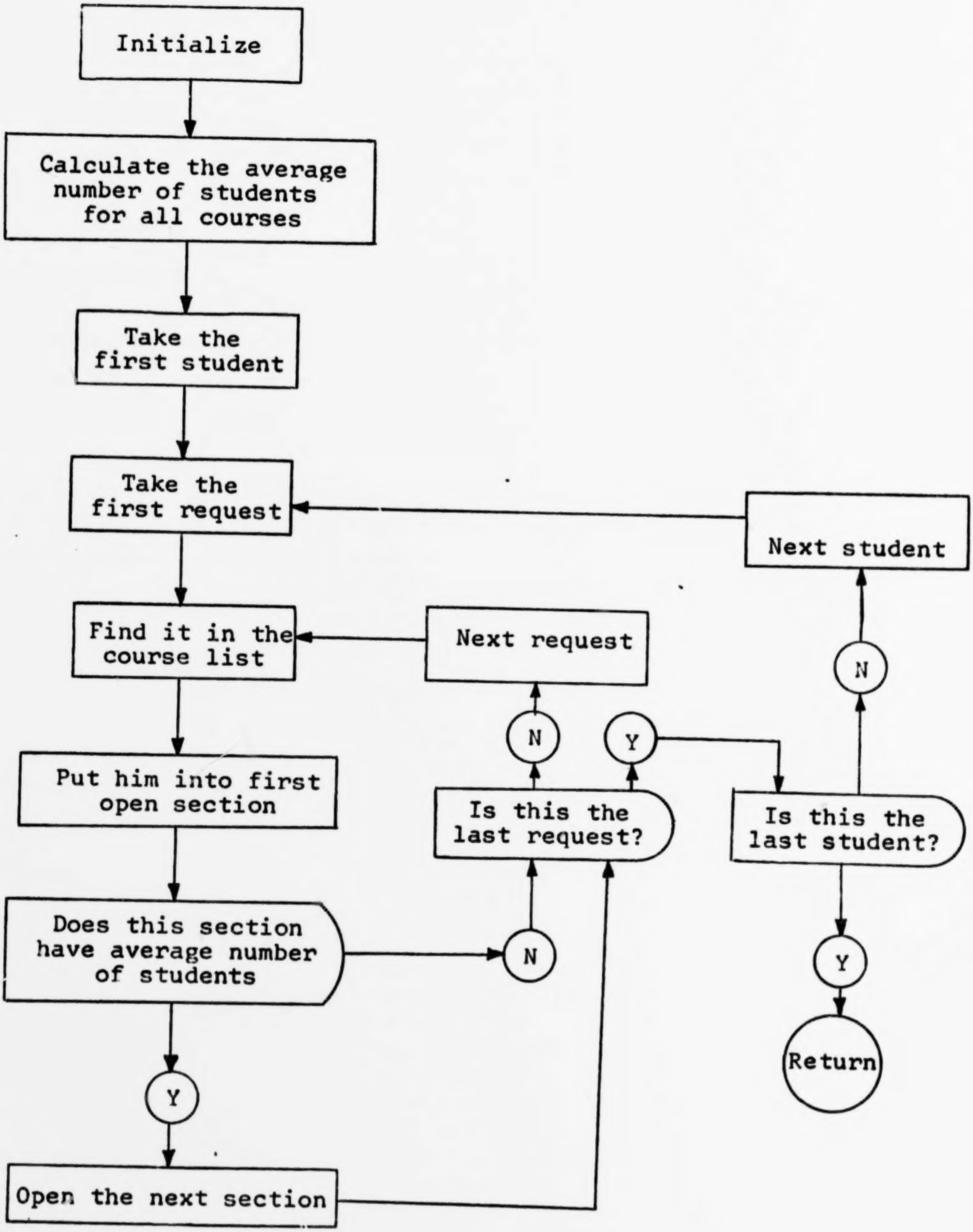

Initialize
Calculate the average number of students for all courses
Take the first student
Take the first request
Next student
Find it in the course list
Next request
N
Y
N
Put him into first open section
Is this the last request?
Is this the last student?
Does this section have average number of students
N
Y
Y
Return
Open the next section

Figure 3

SUB-ROUTINE TWOP (ASSIGNMENT ROUTINE NO. 2)

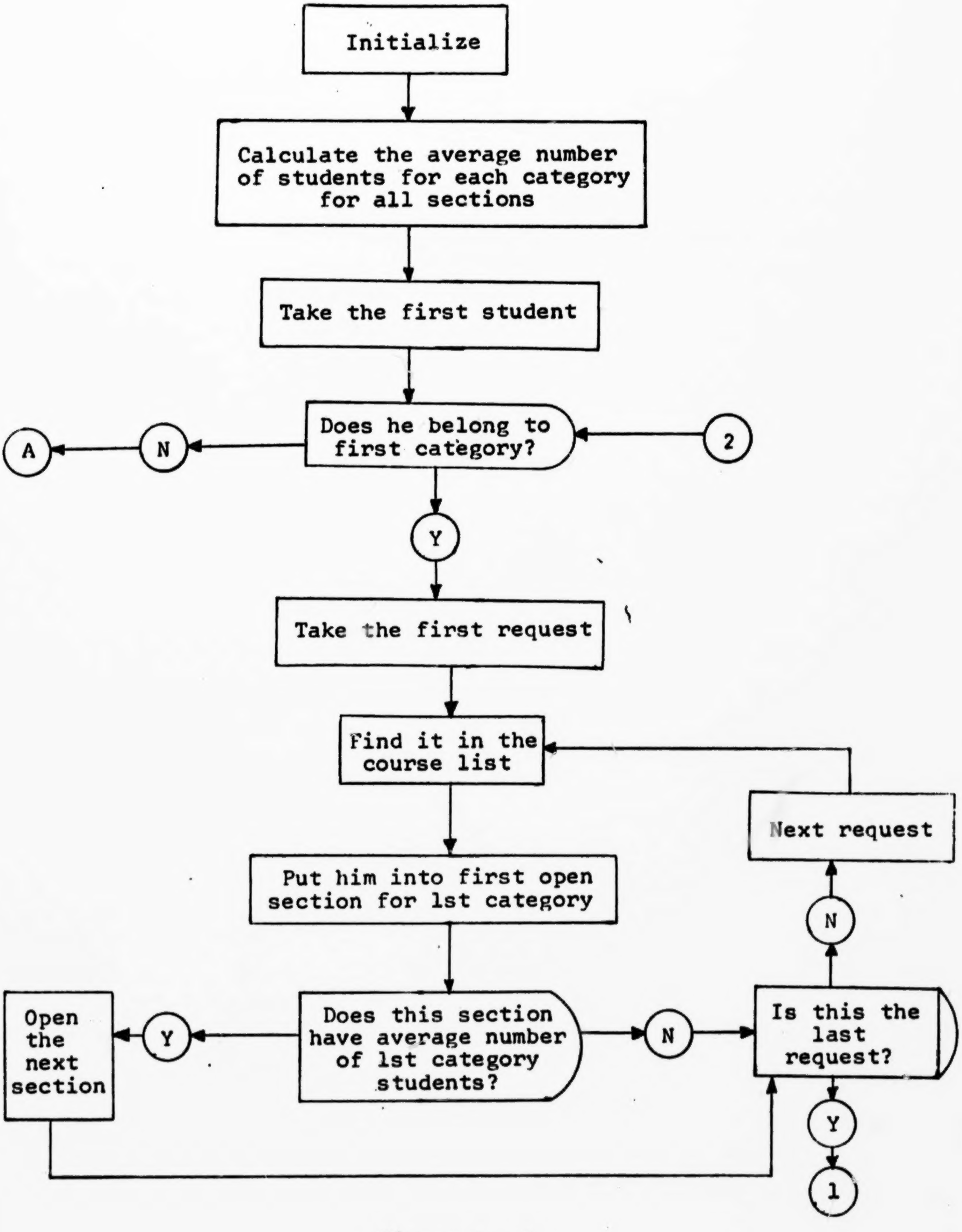

Initialize
Calculate the average number of students for each category for all sections
Take the first student
Does he belong to first category?
A
N
2
Y
Take the first request
Find it in the course list
Next request
Put him into first open section for 1st category
N
Open the next section
Y
Does this section have average number of 1st category students?
N
Is this the last request?
Y
1

Figure 4 - A

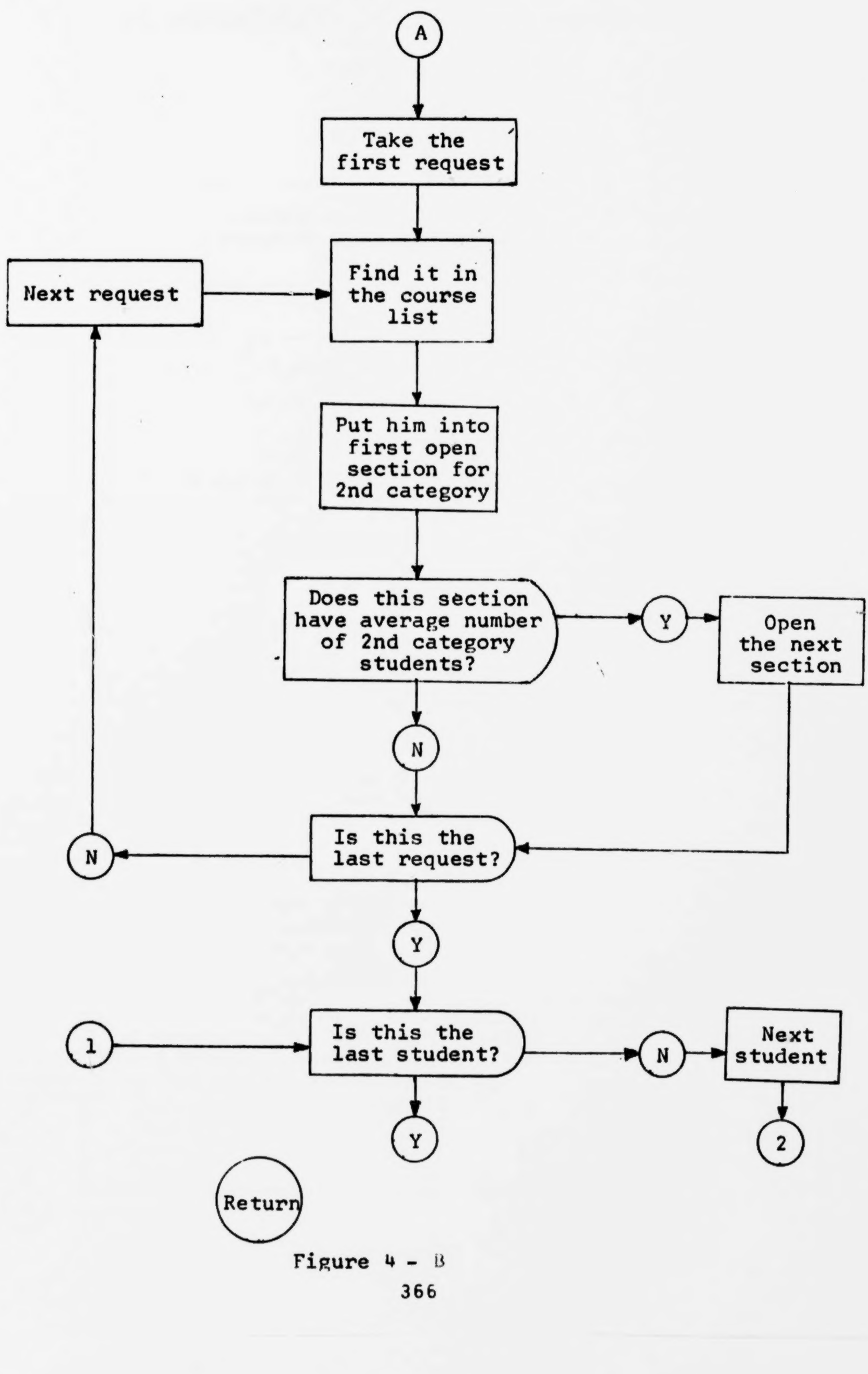

A
Take the first request
Next request
Find it in the course list
Put him into first open section for 2nd category
Does this section have average number of 2nd category students?
Y
Open the next section
N
Is this the last request?
N
Y
1
Is this the last student?
N
Next student
Y
2
Return

Figure 4 - B

SUB-ROUTINE SORT (ORDERING ROUTINE)

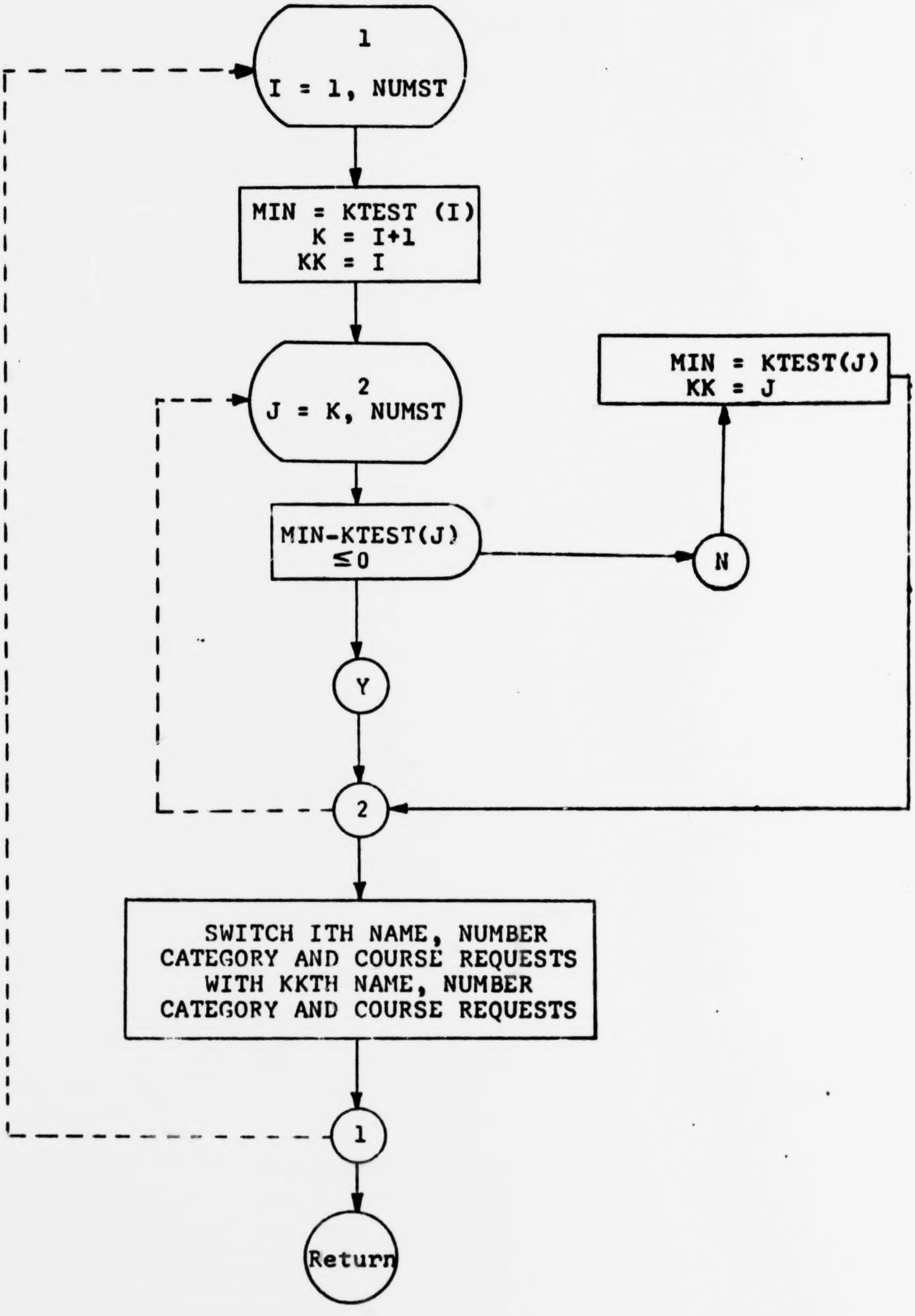

1
I = 1, NUMST
MIN = KTEST (I)
K = I+1
KK = I
2
J = K, NUMST
MIN = KTEST(J)
KK = J
MIN-KTEST(J) ≤0
N
Y
2
SWITCH ITH NAME, NUMBER CATEGORY AND COURSE REQUESTS WITH KKTH NAME, NUMBER CATEGORY AND COURSE REQUESTS
1
Return

Figure 5

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Sub-routine EXECT and MAIN PROGRAM Figure 13 and Figure 14 are used for monitoring and input/output purposes.

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 6

```
      SUBROUTINE ONE
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,NTYPE,NORDR
      DO 100 I=1,200
      KCORS(I)=0
  100 NUM1(I)=0
      DO 105 J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  105 NUM1(J)=NUM1(J)+1
      K=NUMRQ
      DO 130 I=2,NUMST
      DO 130 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)110,130,115
  110 STOP 9599
  115 DO 120 L=1,K
      IF(KCORS(L)-KTEMP)120,125,120
  120 CONTINUE
      K=K+1
      KCORS(K)=KTEMP
      NUM1(K)=NUM1(K)+1
      GO TO 130
  125 NUM1(L)=NUM1(L)+1
  130 CONTINUE
      NOCOR=K
      PRINT 21
   21 FORMAT(/37H COURSE NUMBER    NUMBER OF REQUESTS  /)
      PRINT 5,(KCORS(I),NUM1(I),I=1,NOCOR)
    5 FORMAT(2I10)
      CALL ONEP
      RETURN
      END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 7 - A

```
      SUBROUTINE TWO
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(100),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,NOUTR,NTSEC,NTYPE,NORDR
      DO -100 I=1,200
      KCORS(I)=0
      NUM1(I)=0
  100 NUM2(I)=0
      IF(KATGR(1))101,105,115
  101 STOP 9699
  105 DO 110J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  110 NUM1(J)=NUM1(J)+1
      GO TO 125
  115 DO 120 J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  120 NUM2(J)=NUM2(J)+1
  125 K=NUMRQ
      DO 190 I=2,NUMST
      IF(KATGR(I))101,130,160
  130 DO 150 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)131,150,135
  131 STOP 9599
  135 DO 140 L=1,K
      IF(KCORS(L)-KTEMP)140,145,140
  140 CONTINUE
      K=K+1
      KCORS(K)=KTEMP
      NUM1(K)=NUM1(K)+1
      GO TO 150
  145 NUM1(L)=NUM1(L)+1
  150 CONTINUE
      GO TO 190
  160 DO 180 J=1,NUMRQ
      IF(KSREQ(I,J))161,180,165
  161 STOP 9599
  165 KTEMP=KSREQ(I,J)
      DO 170 L=1,K
      IF(KCORS(L)-KTEMP)170,175,170
  170 CONTINUE
      K=K+1
      KCORS(K)=KSREQ(I,J)
      NUM2(K)=NUM2(K)+1
      GO TO 180
  175 NUM2(L)=NUM2(L)+1
  180 CONTINUE
  190 CONTINUE
      NOCOR=K
      DO 200 I=1,NOCOR
  200 NUTOT(I)=NUM1(I)+NUM2(I)
      PRINT 23
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 7 - B

```
23 FORMAT (/69H COURSE NUMBER NO. OF SECTIONS TOTAL REQUESTS  1ST GRO
  1UP   2ND GROUP  //)
   PRINT 5,(KCORS(I),KNSEC(I),NUTOT(I),NUM1(I),NUM2(I),I=1,NOCOR)
 5 FORMAT (I6,I12,I19,I15,I10)
   CALL TWOP
   RETURN
   END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 8 - A

```
    . SUBROUTINE THREE
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MCNTR,NTSEC,NTYPE,NORDR
      DO 100 I=1,200
      KCORS(I)=0
      NUM1(I)=0
      NUM2(I)=0
  100 NUM3(I)=0
      IF(KATGR(1)-1)105,115,125
  105 DO 110 J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  110 NUM1(J)=NUM1(J)+1
      GO TO 135
  115 DO 120 J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  120 NUM2(J)=NUM2(J)+1
      GO TO 135
  125 DO 130 J=1,NUMRQ
      KCORS(J)=KSREQ(1,J)
  130 NUM3(J)=NUM3(J)+1
  135 K=NUMRQ
      DO 220 I=2,NUMST
      IF(KATGR(I)-1)140,165,190
  140 DO 160 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)141,160,145
  141 STOP 9599
  145 DO 150 L=1,K
      IF(KCORS(L)-KTEMP)150,155,150
  150 CONTINUE
      K=K+1
      KCORS(K)=KTEMP
      NUM1(K)=NUM1(K) +1
      GO TO 160
  155 NUM1(L)=NUM1(L)+1
  160 CONTINUE
      GO TO 220
  165 DO 185 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)166,185,170
  166 STOP 9599
  170 DO 175 L=1,K
      IF(KCORS(L)-KTEMP)175,180,175
  175 CONTINUE
      K=K+1
      KCORS(K)=KTEMP
      NUM2(K)=NUM2(K)+1
      GO TO 185
  180 NUM2(L)=NUM2(L)+1
  185 CONTINUE
      GO TO 220
```

```
190 DO 210 J=1,NUMRQ
    KTEMP=KSREQ(I,J)
    IF(KTEMP)191,210,195
191 STOP 9599
195 DO 200 L=1,K
    IF(KCORS(L)-KTEMP)200,205,200
200 CONTINUE
    K=K+1
    KCORS(K)=KTEMP
    NUM3(K)=NUM3(K)+1
    GO TO 210
205 NUM3(L)=NUM3(L)+1
210 CONTINUE
220 CONTINUE
    NOCOR=K
    DO 225 I=1,NOCOR
225 NUTOT(I)=NUM1(I)+NUM2(I)+NUM3(I)
    PRINT 24
 24 FORMAT (/82H COURSE NUMBER  NO. OF SECTIONS  TOTAL REQUESTS  1ST
   1GROUP  2ND GROUP  3RD GROUP  //)
    PRINT 5,(KCORS(I),KNSEC(I),NUTOT(I),NUM1(I),NUM2(I),NUM3(I),
   1I=1,NOCOR)
  5 FORMAT (I8,I17,I16,I14,2I11)
    CALL THREP
    RETURN
    END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 8 - B

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 9

```
      SUBROUTINE OVER
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,NTYPE,NORDR
      DO 200 I=1,NOCOR
  200 KGRP1(I)=FLOTF(NUM1(I))/FLOTF(KNSEC(I))+0.9
      DO 100 I=1,NOCOR
      KSEC1(I)=1
  100 KTRC1(I)=0
      DO 125 I=1,NUMST
      DO 125 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)111,125,105
  105 DO 110 K=1,NOCOR
      IF(KCORS(K)-KTEMP)110,115,110
  110 CONTINUE
      STOP 9899
  111 STOP 9699
  115 KSREQ(I,J)=100*KTEMP+KSEC1(K)
      KTRC1(K)=KTRC1(K)+1
      IF(KTRC1(K)-KGRP1(K))125,120,120
  120 KSEC1(K)=KSEC1(K)+1
      KTRC1(K)=0
  125 CONTINUE
      PRINT 22
   22 FORMAT(/68H STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS
     1ARE ASSIGNED  //)
      DO 9 I=1,NUMST
    9 PRINT1,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),(KSREQ(I,J),J=1,
     1NUMRQ )
    1 FORMAT (3A5 ,I5,4X,10I8 )
      RETURN
      END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 10

```
      SUBROUTINE TWOP
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,MTYPE,NORDR
      DO 200 I=1,NOCOR
      KGRP1(I)=FLOTF(NUM1(I))/FLOTF(KNSEC(I))+0.9
  200 KGRP2(I)=FLOTF(NUM2(I))/FLOTF(KNSEC(I))+0.9
      DO 105 I=1,NOCOR
      KSEC1(I)=1
      KSEC2(I)=1
      KTRC1(I)=0
  105 KTRC2(I)=0
      DO 160 I=1,NUMST
      IF(KATGR(I))106,110,135
  106 STOP 9699
  110 DO 130 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)111,130,113
  113 DO 115 K=1,NOCOR
      IF(KCORS(K)-KTEMP)115,120,115
  115 CONTINUE
      STOP 9899
  111 STOP 9699
  120 KSREQ(I,J)=100*KTEMP+KSEC1(K)
      KTRC1(K)=KTRC1(K)+1
      IF(KTRC1(K)-KGRP1(K))130,125,125
  125 KSEC1(K)=KSEC1(K)+1
      KTRC1(K)=0
  130 CONTINUE
      GO TO 160
  135 DO 155 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)141,155,138
  138 DO 140 K=1,NOCOR
      IF(KCORS(K)-KTEMP)140,145,140
  140 CONTINUE
      STOP 9899
  141 STOP 9699
  145 KSREQ(I,J)=100*KTEMP+KSEC2(K)
      KTRC2(K)=KTRC2(K)+1
      IF(KTRC2(K)-KGRP2(K))155,150,150
  150 KSEC2(K)=KSEC2(K)+1
      KTRC2(K)=0
  155 CONTINUE
  160 CONTINUE
      PRINT 22
   22 FORMAT(/68H STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS
     1ARE ASSIGNED  //)
      DO 9 I=1,NUMST
    9 PRINT10,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),(KSREQ(I,J),J=1,
     1NUMRQ )
   10 FORMAT (3A5 ,I5,4X,10I8 )
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 11 - A

```
      SUBROUTINE THREP
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(100),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,NTYPE,NORDR
      DO 100 I=1,NOCOR
      KGRP1(I)=FLOTF(NUM1(I))/FLOTF(KNSEC(I))+0.9
      KGRP2(I)=FLOTF(NUM2(I))/FLOTF(KNSEC(I))+0.9
  100 KGRP3(I)=FLOTF(NUM3(I))/FLOTF(KNSEC(I))+0.9
      DO 105 I=1,NOCOR
      KSEC1(I)=1
      KSEC2(I)=1
      KSEC3(I)=1
      KTRC1(I)=0
      KTRC2(I)=0
  105 KTRC3(I)=0
      DO 190 I=1,NUMST
      IF(KATGR(I)-1)110,135,160
  110 DO 130 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)116,130,112
  112 DO 115 K=1,NOCOR
      IF(KCORS(K)-KTEMP)115,120,115
  115 CONTINUE
      STOP 9899
  116 STOP 9699
  120 KSREQ(I,J)=100*KTEMP+KSEC1(K)
      KTRC1(K)=KTRC1(K)+1
      IF(KTRC1(K)-KGRP1(K))130,125,125
  125 KSEC1(K)=KSEC1(K)+1
      KTRC1(K)=0
  130 CONTINUE
      GO TO 190
  135 DO 155 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)141,155,137
  137 DO 140 K=1,NOCOR
      IF(KCORS(K)-KTEMP)140,145,140
  140 CONTINUE
      STOP 9899
  141 STOP 9699
  145 KSREQ(I,J)=100*KTEMP+KSEC2(K)
      KTRC2(K)=KTRC2(K)+1
      IF(KTRC2(K)-KGRP2(K))155,150,150
  150 KSEC2(K)=KSEC2(K)+1
      KTRC2(K)=0
  155 CONTINUE
      GO TO 190
  160 DO 180 J=1,NUMRQ
      KTEMP=KSREQ(I,J)
      IF(KTEMP)166,180,162
  162 DO 165 K=1,NOCOR
      IF(KCORS(K)-KTEMP)165,170,165
```

```
165 CONTINUE
    STOP 9899
166 STOP 9699
170 KSREQ(I,J)=100*KTEMP+KSEC3(K)
    KTRC3(K)=KTRC3(K)+1
    IF(KTRC3(K)-KGRP3(K))180,175,175
175 KSEC3(K)=KSEC3(K)+1
    KTRC3(K)=0
180 CONTINUE
190 CONTINUE
    PRINT 22
 22 FORMAT(/68H STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS
   1ARE ASSIGNED  //)
    DO 9 I=1,NUMST
  9 PRINT10,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),(KSREQ(I,J),J=1,
   1NUMRO )
 10 FORMAT (3A5 ,I5,4X,10I8 )
    RETURN
    END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 11 - B

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 12

```
      SUBROUTINE SORT
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,NTYPE,NORDR
      DIMENSION KEMP(10)
      KKK=NUMST-1
      DO 130 I=1,KKK
      MIN=KTEST(I)
      K=I+1
      KK=I
      DO 110 J=K,NUMST
      IF(MIN-KTEST(J))110,110,105
  105 MIN=KTEST(J)
      KK=J
  110 CONTINUE
      KEMP1=NAME1(I)
      KEMP2=NAME2(I)
      KEMP3=NAME3(I)
      KEMP4=KSNUM(I)
      KEMP5=KTEST(I)
      KEMP6=KATGR(I)
      DO 115 L=1,NUMRQ
  115 KEMP(L)=KSREQ(I,L)
      NAME1(I)=NAME1(KK)
      NAME2(I)=NAME2(KK)
      NAME3(I)=NAME3(KK)
      KSNUM(I)=KSNUM(KK)
      KTEST(I)=KTEST(KK)
      KATGR(I)=KATGR(KK)
      DO 120 L=1,NUMRQ
  120 KSREQ(I,L)=KSREQ(KK,L)
      NAME1(KK)=KEMP1
      NAME2(KK)=KEMP2
      NAME3(KK)=KEMP3
      KSNUM(KK)=KEMP4
      KTEST(KK)=KEMP5
      KATGR(KK)=KEMP6
      DO 125 L=1,NUMRQ
  125 KSREQ(KK,L)=KEMP(L)
  130 CONTINUE
      PRINT 20
   20 FORMAT(//50H STUDENTS ARE ORDERED ACCORDING TO THE GIVEN TEST SCOR
     1ES //)
      DO 24 I=1,NUMST
   24 PRINT 25,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),KTEST(I),
     1KATGR(I),(KSREQ(I,J),J=1,NUMRQ)
   25 FORMAT (  3A5,I6,2I13 ,11X,10I6 )
      RETURN
      END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

Figure 13

```
      SUBROUTINE EXECT
      DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
      COMMON NAME1,NAME2,NAME3,KTEST
      DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
      COMMON KSREQ,KODE1,KATGR,KSNUM
      DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
     1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
     2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
      COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
     1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
      COMMON NUMST,NUMRQ,NOCOR,MONTR,NTSEC,NTYPE,NORDR
      IF(NORDR)100,110,105
  100 STOP 9699
  105 CALL SQRT
  110 IF(NTYPE)100,115,120
  115 STOP 9499
  120 GO TO (125,130,135),NTYPE
  125 CALL ONE
      RETURN
  130 CALL TWO
      RETURN
  135 CALL THREE
      RETURN
      END
```

PROGRAMS FOR ASSIGNMENT OF STUDENTS TO COURSE SECTIONS

MAIN PROGRAM

Figure 14

```
   DIMENSION NAME1(100),NAME2(100),NAME3(100),KTEST(100)
   COMMON NAME1,NAME2,NAME3,KTEST
   DIMENSION KSREQ(100,10),KODE1(1000),KATGR(100),KSNUM(100)
   COMMON KSREQ,KODE1,KATGR,KSNUM
   DIMENSION KCORS(200),KNSEC(200),NUTOT(200) ,
  1NUM1(200),NUM2(200),NUM3(200),KGRP1(200),KGRP2(200),KGRP3(200),
  2KSEC1(200),KSEC2(200),KSEC3(200),KTRC1(200),KTRC2(200),KTRC3(200)
   COMMON KCORS,KNSEC,NUTOT,NUM1,NUM2,NUM3,
  1KGRP1,KGRP2,KGRP3,KSEC1,KSEC2,KSEC3,KTRC1,KTRC2,KTRC3
   COMMON NUMST,NUMRQ,NOCOR,NONTR,NTSEC,NTYPE,NORDR
   READ 5,NTYPE,NORDR
   READ 5,(KNSEC(I),I=1,11)
 5 FORMAT (16I5)
   READ 5,NUMST,NUMRQ
   DO 9 I=1,NUMST
 9 READ 10,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),KTEST(I),KATGR(I),
  1(KSREQ(I,J),J=1,NUMRQ)
   IF(NORDR)59,60,65
59 STOP 90
60 GO TO (30,35,35),NTYPE
65 GO TO (40,45,45),NTYPE
30 PRINT 31
31 FORMAT(//15X,41HBALANCING CRITERION   (SECOND CRITERION)  )
   GO TO 50
35 PRINT 36
36 FORMAT (//15X,55HHETEROGENEOUS BALANCING CRITERION   (FOURTH CRITE
  1RION)  )
   GO TO 50
40 PRINT 41
41 FORMAT ( 15X,66HHOMOGENEOUS GROUPING WITH BALANCING CRITERION   (F
  1IFTH CRITERION)  )
   GO TO 50
45 PRINT 46
46 FORMAT ( 15X,81HHOMOGENEOUS  GROUPING WITH HETEROGENEOUS BALANCING
  1 CRITERION   (SIXTH CRITERION)  )
50 PRINT 20
20 FORMAT (//81H STUDENT NAME   STUD. NO.   TEST SCORE   CATEGORY
  1    STUDENT COURSE REQUESTS   //)
   DO 19 I=1,NUMST
19 PRINT 11,NAME1(I),NAME2(I),NAME3(I),KSNUM(I),KTEST(I),KATGR(I),
  1(KSREQ(I,J),J=1,NUMRQ)
10 FORMAT(3A5,13I5)
11 FORMAT (  3A5,I6,2I13 ,11X,10I6 )
   CALL EXECT
   PRINT 47
47 FORMAT (C 2)
   STOP
   END
```

```
CURRAN  W. K.      1  391     0    18    26    12    24    14    16    22
BANEY  E. F.       2  400     1    18    26    12    24    14    16    22
ODORISO  R. S.     3  385     1    18    26    20    24    14    16    30
JOHNSON  A. B.     4  394     0    18    26    12    28    14    30    16
HOWARD  K. K.      5  395     0    18    26    20    28    14    22    30
EASTON  C. D.      6  402     1    18    26    12    24    14    16    22
PAULUS  G. F.      7  388     1    18    10    20    24    14    16    30
ROSS  M.  M.       8  380     1    16    10    20    24    14    22    30
MILLER   D.C.      9  403     0    18    26    20    28    16    22    30
SMITH  F. G.      10  390     1    18    10    20    28
```

SAMPLE PROBLEM DATA

Figure 15

```
                  BALANCING CRITERION   (SECOND CRITERION)

STUDENT NAME   STUD. NO.   TEST SCORE   CATEGORY        STUDENT COURSE REQUESTS


CURRAN  W. K.      1          391          0            18    26    12    24    14    16    22
BANEY  E. F.       2          400          1            18    26    12    24    14    16    22
UDORISO  R. S.     3          385          1            18    26    20    24    14    16    30
JOHNSON  A. B.     4          394          0            18    26    12    28    14    30    16
HOWARD  K. K.      5          395          0            18    26    20    28    14    22    30
EASTON  C. D.      6          402          1            18    26    12    24    14    16    22
PAULUS  G. F.      7          388          1            18    10    20    24    14    16    30
ROSS  M.  N.       8          380          1            18    10    20    24    14    22    30
MILLER   D.C.      9          403          0            18    26    20    28    16    22    30
SMITH  F. G.      10          390          1            18    10    20    28     0     0     0
COURSE NUMBER   NUMBER OF REQUESTS

      18          10
      26           7
      12           4
      24           6
      14           8
      16           7
      22           6
      20           6
      30           6
      28           4
      10           3
STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS ARE ASSIGNED


CURRAN  W. K.      1          1801     2601     1201     2401     1401     1601     2201
BANEY  E. F.       2          1801     2601     1201     2401     1401     1601     2201
UDORISO  R. S.     3          1801     2601     2001     2401     1401     1601     3001
JOHNSON  A. B.     4          1801     2601     1202     2801     1401     3001     1601
HOWARD  K. K.      5          1801     2602     2001     2801     1401     2201     3002
EASTON  C. D.      6          1801     2602     1202     2402     1401     1601     2201
PAULUS  G. F.      7          1801     1001     2001     2402     1401     1601     3002
ROSS  M.  N.       8          1801     1001     2002     2402     1401     2201     3003
MILLER   D.C.      9          1801     2602     2002     2801     1601     2201     3003
SMITH  F. G.      10          1801     1002     2002     2801        0        0        0
```

OUTPUT

Figure 16

OUTPUT

Figure 17

```
                HETEROGENEOUS BALANCING CRITERION   (FOURTH CRITERION)

STUDENT NAME   STUD. NO.   TEST SCORE   CATEGORY       STUDENT COURSE REQUESTS


CURRAN  W. K.      1           391          0            18    26    12    24    14    16    22
BANEY  E. F.       2           400          1            18    26    12    24    14    16    22
ODORISO  R. S.     3           385          1            18    26    20    24    14    16    30
JOHNSON  A. B.     4           394          0            18    26    12    28    14    30    16
HOWARD  K. K.      5           395          0            18    26    20    28    14    22    30
EASTON  C. D.      6           402          1            18    26    12    24    14    16    22
PAULUS  G. F.      7           388          1            18    10    20    24    14    16    30
ROSS  M. N.        8           380          1            18    10    20    24    14    22    30
MILLER  D.C.       9           403          0            18    26    20    28    16    22    30
SMITH  F. G.      10           390          1            18    10    20    28     0     0     0
COURSE NUMBER NO. OF SECTIONS TOTAL REQUESTS  1ST GROUP   2ND GROUP


     18             1              10           4          6
     26             2               7           4          3
     12             3               4           2          2
     24             2               6           1          5
     14             1               8           3          5
     16             1               7           3          4
     22             1               6           3          3
     20             2               6           2          4
     30             3               6           3          3
     28             1               4           3          1
     10             2               3           0          3
STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS ARE ASSIGNED


CURRAN  W. K.      1          1801    2601    1201    2401    1401    1601    2201
BANEY  E. F.       2          1801    2601    1201    2401    1401    1601    2201
ODORISO  R. S.     3          1801    2601    2001    2401    1401    1601    3001
JOHNSON  A. B.     4          1801    2601    1202    2801    1401    3001    1601
HOWARD  K. K.      5          1801    2602    2001    2801    1401    2201    3002
EASTON  C. D.      6          1801    2602    1202    2401    1401    1601    2201
PAULUS  G. F.      7          1801    1001    2001    2402    1401    1601    3002
ROSS  M. N.        8          1801    1001    2002    2402    1401    2201    3003
MILLER  D.C.       9          1801    2602    2002    2801    1601    2201    3003
SMITH  F. G.      10          1801    1002    2002    2801       0       0       0
```

OUTPUT

Figure 18

```
               HOMOGENEOUS GROUPING WITH BALANCING CRITERION   (FIFTH CRITERION)

     STUDENT NAME   STUD. NO.   TEST SCORE   CATEGORY        STUDENT COURSE REQUESTS


     CURRAN  W. K.      1          391          0             18    26    12    24    14    16    22
     BANEY  E. F.       2          400          1             18    26    12    24    14    16    22
     ODORISO  R. S.     3          385          1             18    26    20    24    14    16    30
     JOHNSON  A. B.     4          394          0             18    26    12    28    14    30    16
     HOWARD  K. K.      5          395          0             18    26    20    28    14    22    30
     EASTON  C. D.      6          402          1             18    26    12    24    14    16    22
     PAULUS  G. F.      7          388          1             18    10    20    24    14    16    30
     ROSS  M. N.        8          380          1             18    10    20    24    14    22    30
     MILLER   D.C.      9          403          0             18    26    20    28    16    22    30
     SMITH  F. G.      10          390          1             18    10    20    28     0     0     0

     STUDENTS ARE ORDERED ACCORDING TO THE GIVEN TEST SCORES


     ROSS  M. N.        8          380          1             18    10    20    24    14    22    30
     ODORISO  R. S.     3          385          1             18    26    20    24    14    16    30
     PAULUS  G. F.      7          388          1             18    10    20    24    14    16    30
     SMITH  F. G.      10          390          1             18    10    20    28     0     0     0
     CURRAN  W. K.      1          391          0             18    26    12    24    14    16    22
     JOHNSON  A. B.     4          394          0             18    26    12    28    14    30    16
     HOWARD  K. K.      5          395          0             18    26    20    28    14    22    30
     BANEY  E. F.       2          400          1             18    26    12    24    14    16    22
     EASTON  C. D.      6          402          1             18    26    12    24    14    16    22
     MILLER   D.C.      9          403          0             18    26    20    28    16    22    30
     COURSE NUMBER   NUMBER OF REQUESTS

           18          10
           10           3
           20           6
           24           6
           14           8
           22           6
           30           6
           26           7
           16           7
           28           4
           12           4

     STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS ARE ASSIGNED


     ROSS  M. N.        8        1801    1001    2001    2401    1401    2201    3001
     ODORISO  R. S.     3        1801    2601    2001    2401    1401    1601    3001
     PAULUS  G. F.      7        1801    1001    2002    2401    1401    1601    3001
     SMITH  F. G.      10        1801    1002    2002    2801       0       0       0
     CURRAN  W. K.      1        1801    2601    1201    2402    1401    1601    2201
     JOHNSON  A. B.     4        1801    2601    1201    2801    1401    3001    1602
     HOWARD  K. K.      5        1801    2601    2003    2801    1401    2201    3001
     BANEY  E. F.       2        1801    2602    1202    2402    1401    1602    2201
     EASTON  C. D.      6        1801    2602    1202    2402    1401    1602    2201
     MILLER   D.C.      9        1801    2602    2003    2801    1603    2201    3001
```

OUTPUT

Figure 19

```
              HOMOGENEOUS  GROUPING WITH HETEROGENEOUS BALANCING CRITERION.   (SIXTH CRITERION)

STUDENT NAME   STUD. NO.   TEST SCORE   CATEGORY         STUDENT COURSE REQUESTS


CURRAN  W. K.      1          391          0             18    26    12    24    14    16    22
BANEY  E. F.       2          400          1             18    26    12    24    14    16    22
ODORISO  R. S.     3          385          1             18    26    20    24    14    16    30
JOHNSON  A. B.     4          394          0             18    26    12    28    14    30    16
HOWARD  K. K.      5          395          0             18    26    20    28    14    22    30
EASTON  C. D.      6          402          1             18    26    12    24    14    16    22
PAULUS  G. F.      7          388          1             18    10    20    24    14    16    30
ROSS  M.  N.       8          380          1             18    10    20    24    14    22    30
MILLER   D.C.      9          403          0             18    26    20    28    16    22    30
SMITH  F. G.      10          390          1             18    10    20    28     0     0     0

STUDENTS ARE ORDERED ACCORDING TO THE GIVEN TEST SCORES


ROSS  M.  N.       8          380          1             18    10    20    24    14    22    30
ODORISO  R. S.     3          385          1             18    26    20    24    14    16    30
PAULUS  G. F.      7          388          1             18    10    20    24    14    16    30
SMITH  F. G.      10          390          1             18    10    20    28     0     0     0
CURRAN  W. K.      1          391          0             18    26    12    24    14    16    22
JOHNSON  A. B.     4          394          0             18    26    12    28    14    30    16
HOWARD  K. K.      5          395          0             18    26    20    28    14    22    30
BANEY  E. F.       2          400          1             18    26    12    24    14    16    22
EASTON  C. D.      6          402          1             18    26    12    24    14    16    22
MILLER   D.C.      9          403          0             18    26    20    28    16    22    30
COURSE NUMBER NO. OF SECTIONS TOTAL REQUESTS  1ST GROUP   2ND GROUP


     18              1               10            4          6
     10              2                3            0          3
     20              3                6            2          4
     24              2                6            1          5
     14              1                8            3          5
     22              1                6            3          3
     30              1                6            3          3
     26              2                7            4          3
     16              3                7            3          4
     28              1                4            3          1
     12              2                4            2          2
STUDENT NAME STUD. NO.  COURSE SECTIONS THAT STUDENTS ARE ASSIGNED


ROSS  M.  N.       8         1801     1001     2001     2401     1401     2201     3001
ODORISO  R. S.     3         1801     2601     2001     2401     1401     1601     3001
PAULUS  G. F.      7         1801     1001     2002     2401     1401     1601     3001
SMITH  F. G.      10         1801     1002     2002     2801        0        0        0
CURRAN  W. K.      1         1801     2601     1201     2401     1401     1601     2201
JOHNSON  A. B.     4         1801     2601     1202     2801     1401     3001     1602
HOWARD  K. K.      5         1801     2602     2001     2801     1401     2201     3001
BANEY  E. F.       2         1801     2601     1201     2402     1401     1602     2201
EASTON  C. D.      6         1801     2602     1202     2402     1401     1602     2201
MILLER   D.C.      9         1801     2602     2002     2801     1603     2201     3001
```

ROUTINE TO GENERATE COURSE REQUESTS
AND
PROVIDE FILE MAINTENANCE
FOR
STUDENT RECORDS

BY

WAYNE E. BAUGHMAN

INTRODUCTION

An important input into a student scheduling routine is the student request for courses. The description of a system to provide this function follows.

The system is designed basically to generate the course requests as opposed to the conventional method of having the student and/or his advisor determine the courses which the student can and should take. An auxiliary function of the system is to maintain and update student records and print grade reports. The program for the routine used is written in Fortran. It is being run and debugged on an IBM 7070 computer with plans to have the completed package adapted to an IBM 7090.

DESCRIPTION OF THE SYSTEM AND THE SUB-ROUTINES

The program consists of eleven sub-programs and one main or control program. The functions of the main program are to read instructions for the system and to establish the calling sequence of the sub-programs. The total system is shown schematically in Figure 1. The flow diagram for the main program is shown in Figure 2. A listing of the program and sample data are shown in Figure 11. Sample output is shown in Figure 12. A brief description of each of the sub-programs follows:

1. Sub-routine CURIC (Figure 3) - Sub-routine reads the information pertaining to a given curriculum under which a student may be enrolled and writes the

information on a curriculum tape. The input to this sub-routine is the number assigned to this curriculum, the required courses in the curriculum in order of preference, if any, a list of courses (if any) that have prerequisites, and the prerequisites for these courses. In general, this information would be written on tape prior to execution of the other phases of the program.

2. Sub-routine ESTBL (Figure 4) - Sub-routine initiates a master file on tape for a student. The input for this routine consists of any personal information concerning the student (name, student number, address, high school attended, etc.), the curriculum number under which the student is registered, and information (course name, number of credits, grade received) pertaining to the courses which the student has taken up to the time the record is established. This information is compiled into one array which is then written on the master tape. Additional input at this point is a list of courses which the student requests as electives. The electives are also carried in the master file of the student.

3. Sub-routine ADDST (Figure 4) - If a new student enrolls in school after the master tape has already been established, this sub-routine will be used.

4. Sub-routine REVIS (Figure 5) - Sub-routine adds new course information to a student's master file. This routine will be called when a student whose record

is already on the tape has information to add to the record. The two occasions considered are when a student changes his requested electives and when he has completed additional course work.

5. Sub-routine WRITE (Figure 6) - Sub-routine writes on the master tape, the curriculum number, the student's up-to-date record and the most recent request for electives for a particular student.
6. Sub-routine READ (Figure 6) - Sub-routine is identical with WRITE except it reads tape.
7. Sub-routine PRINT (Figure 7) - Sub-routine prints a student's complete record consisting of all personal information contained in the student's file, the courses taken by the student by term, term quality point average, and overall quality point average. The routine when used with the sub-routine REVIS prints the student's record after the record has been updated. When used in conjunction with PRREC (see below), a particular student's record can be printed.
8. Sub-routine PRREC (Figure 7) - Sub-routine locates a particular student's record on the master tape and prints the complete record.
9. Sub-routine DMAND (Figure 8) - Sub-routine generates the demand for courses in the form of course requests by student. The routine considers courses that have been taken by the student, scans the curriculum of the student,and considering required courses and

their prerequisites, if any, writes on tape and/or prints the list of courses that the student can take. If a certain number of courses are specified by the student only these courses will be requested. If the number of courses requested cannot be fulfilled from the curriculum (required courses), then elective courses will be requested. The order the courses will be requested is dictated by the order the courses are listed in the curriculum, with prerequisites considered.

10. Sub-routine GCALC (Figure 9) - This sub-routine is completely internal to the system and is used to determine the total quality points, total credits, term quality point average, and overall quality point average of the student.
11. Sub-routine BINRY (Figure 10) - Also internal to the system, this routine indicates which courses have been passed satisfactorily and which have not. It constructs an array for each student which is carried in the student's master file.

SUMMARY

This aspect of the project is designed to provide the feedback to the total scheduling system in order to operate the complete system with a minimum number of external decisions and interventions. In this context the most important function of this program is to generate the student course requests. Another important aspect of the program is its reporting and file maintenance ability.

FUTURE RESEARCH

Future research in this area of the project will be required for refinement of the system. Additional factors to be considered are the number of course hours requested instead of the number of courses as the program presently does. Also the student should be able to request a given number of required course hours and a given number of elective course hours. The possibility of taking equivalent courses, which will provide more flexibility if the requested course in the curriculum is not offered, must be investigated. Also an advisor may want a student to receive credit for a course which is not specifically required in his curriculum but which is equivalent to the required course. This is not handled in general in the program. Another factor which must be considered is the possibility of co-requisite courses.

Before the program will be satisfactory for long runs, it will be necessary to determine estimated running time of the program. This in turn may put restrictions on the flexibility of the input.

A routine should be written to be used in conjunction with an actual scheduling program, which will initially generate course requests as the program does now, but if a student cannot get a course which he requests, the routine will locate the student's requests and select the next best course.

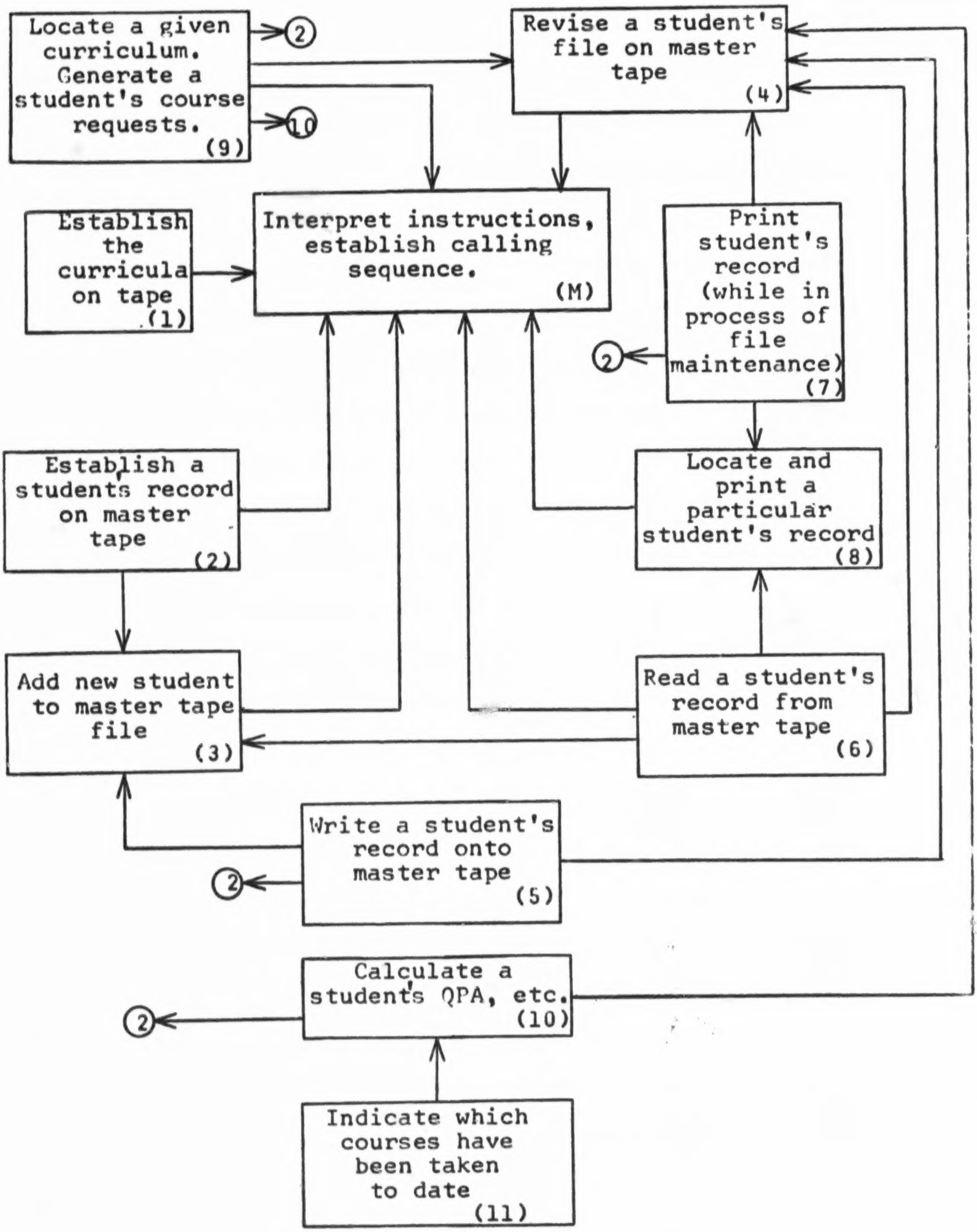

Locate a given curriculum. Generate a student's course requests. (9)
2
10
Revise a student's file on master tape (4)
Establish the curricula on tape (1)
Interpret instructions, establish calling sequence. (M)
Print student's record (while in process of file maintenance) (7)
2
Establish a students record on master tape (2)
Locate and print a particular student's record (8)
Add new student to master tape file (3)
Read a student's record from master tape (6)
Write a student's record onto master tape (5)
2
Calculate a students QPA, etc. (10)
2
Indicate which courses have been taken to date (11)

Figure 1

LEGEND:

M- Main Program
1- Sub-routine CURIC
2- " ESTBL
3- " ADDST
4- " REVIS
5- " WRITE
6- " READ
7- " PRINT
8- " PRREC
9 " DMAND
10- " GCALC
11- " BINRY

In above diagram, the arrow has a special meaning

A ← B

This means B is a sub-routine of A.

Figure 1 continued

MAIN PROGRAM - (M)

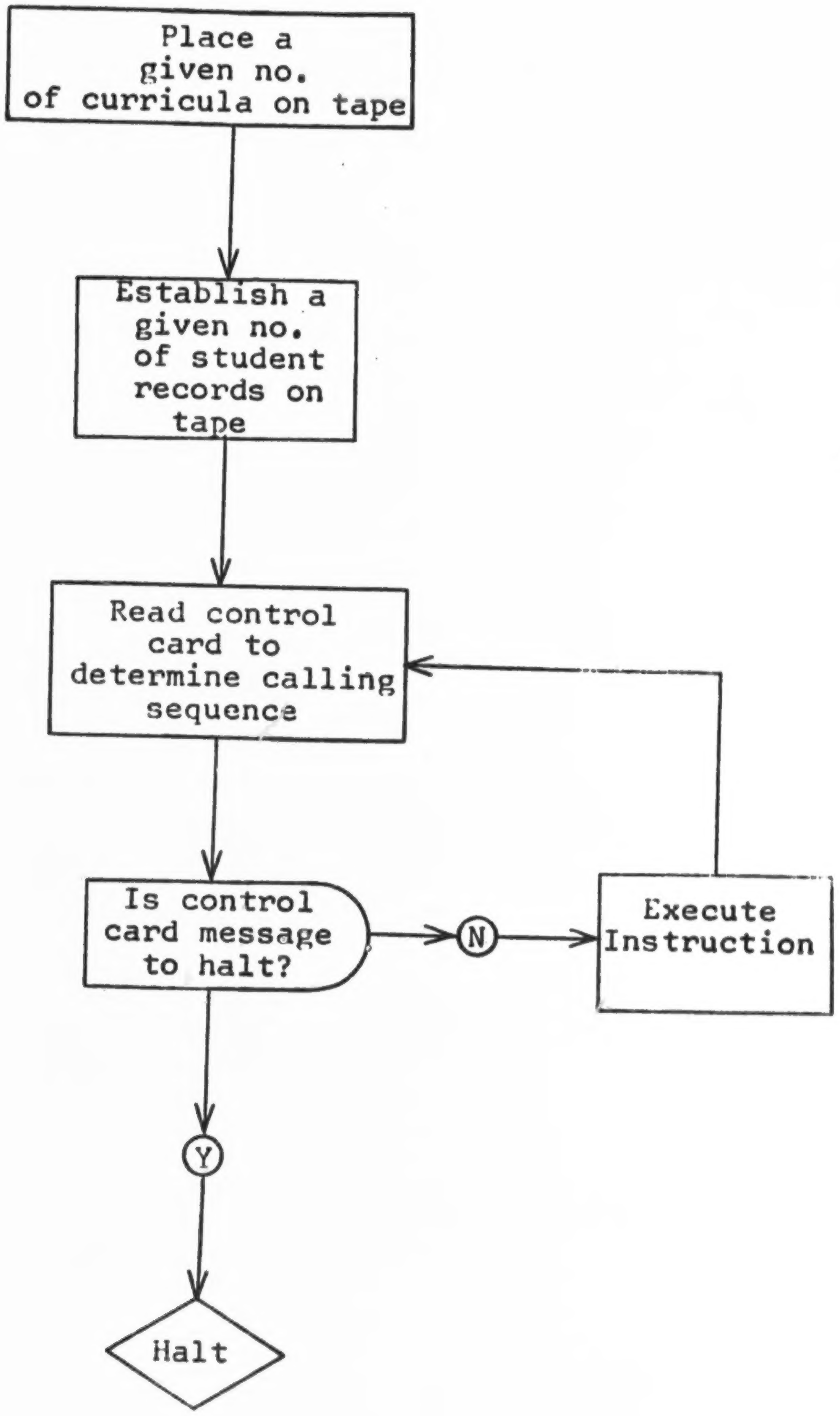

Place a given no. of curricula on tape
Establish a given no. of student records on tape
Read control card to determine calling sequence
Is control card message to halt?
N
Execute Instruction
Y
Halt

Figure 2

SUB-ROUTINE CURIC - (1)

Writes a curriculum on tape

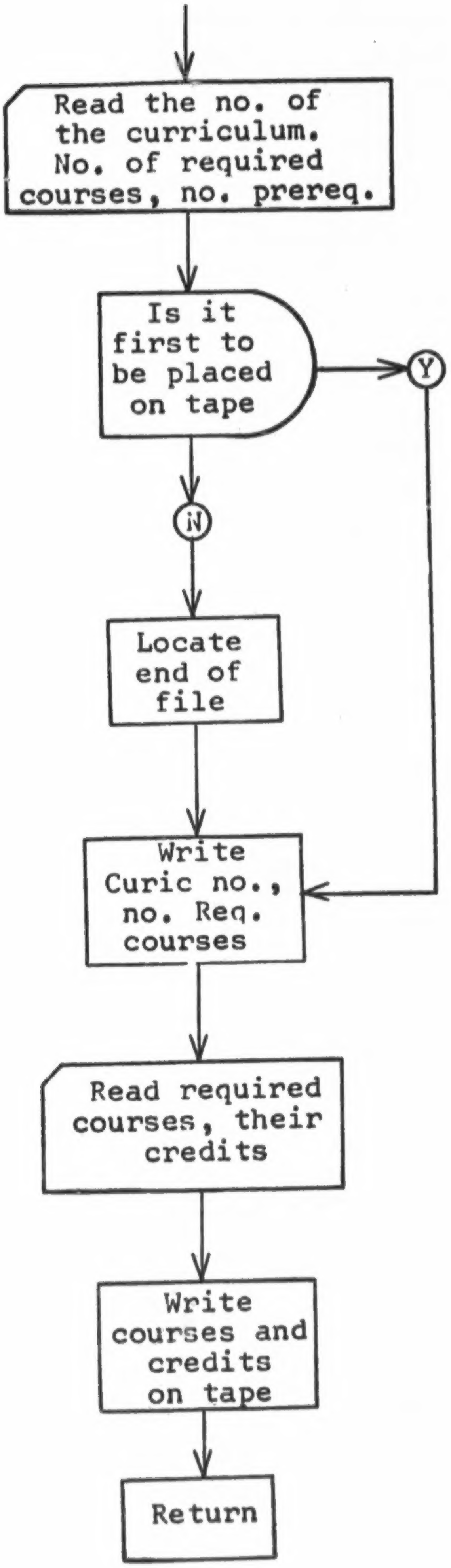

Read the no. of the curriculum. No. of required courses, no. prereq.
Is it first to be placed on tape
Y
N
Locate end of file
Write Curic no., no. Req. courses
Read required courses, their credits
Write courses and credits on tape
Return

Figure 3

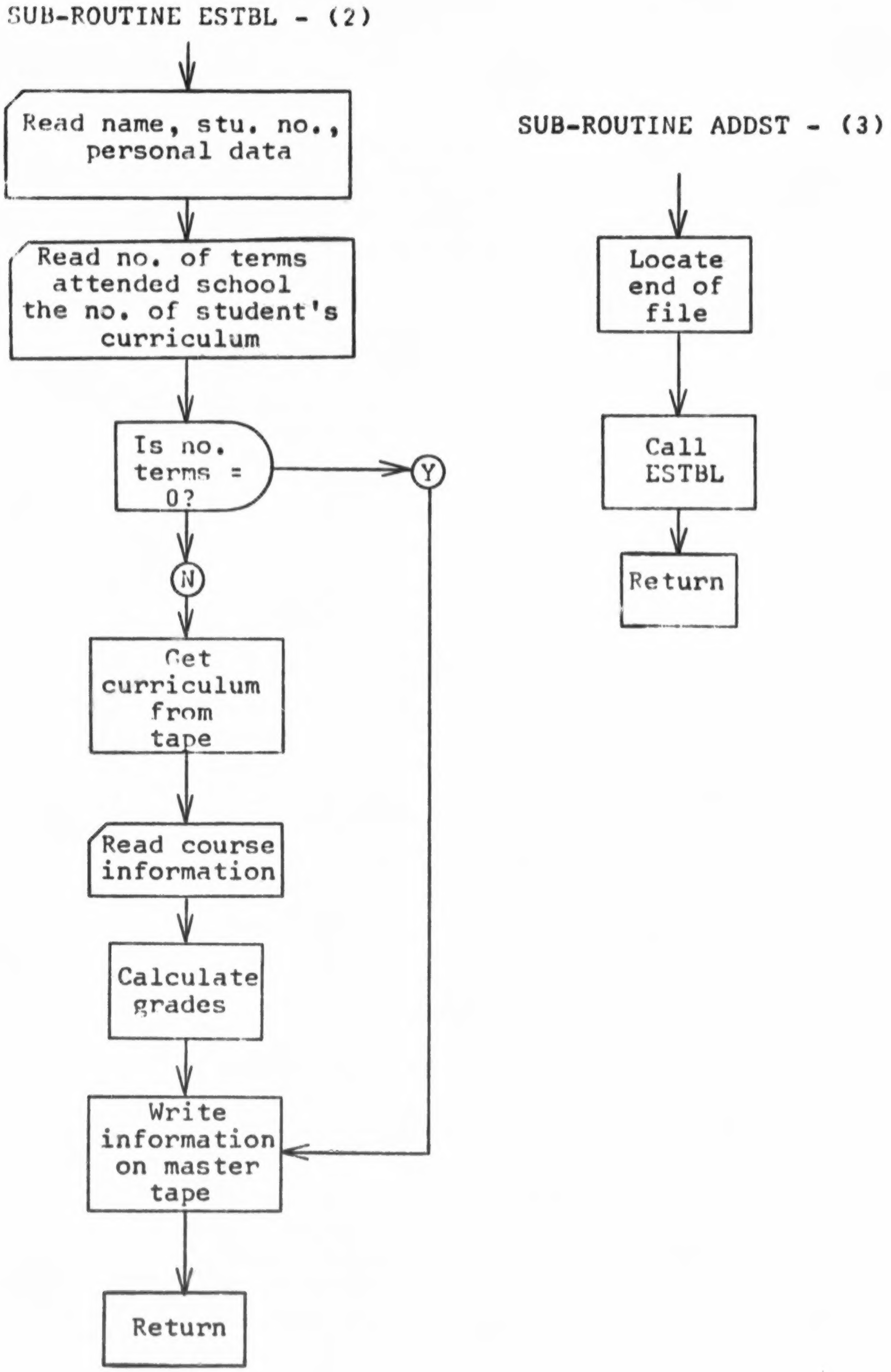

SUB-ROUTINE ESTBL - (2)
Read name, stu. no., personal data
Read no. of terms attended school the no. of student's curriculum
Is no. terms = 0?
Y
N
Get curriculum from tape
Read course information
Calculate grades
Write information on master tape
Return
SUB-ROUTINE ADDST - (3)
Locate end of file
Call ESTBL
Return

Figure 4

SUB-ROUTINE REVIS - (4)

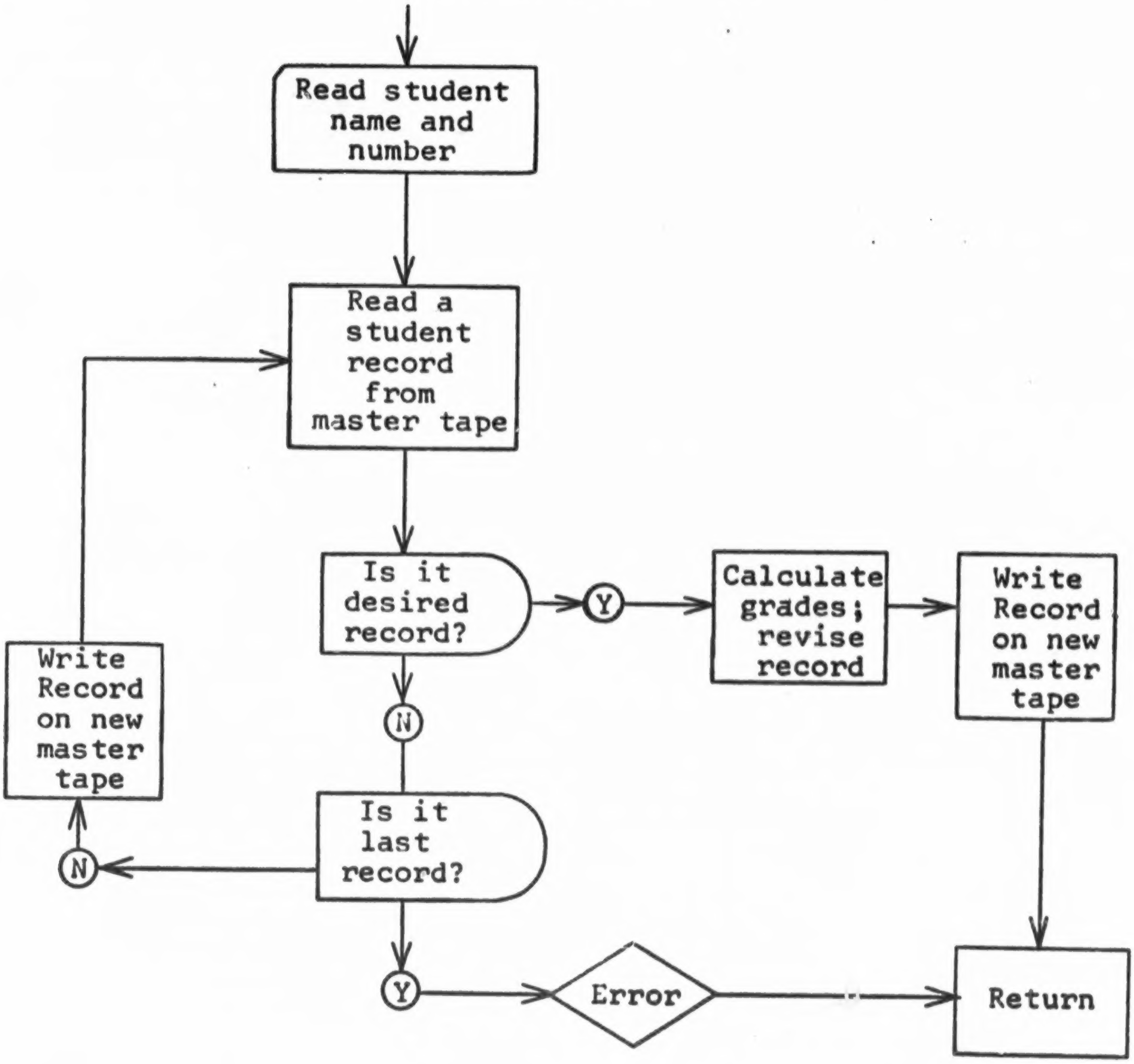
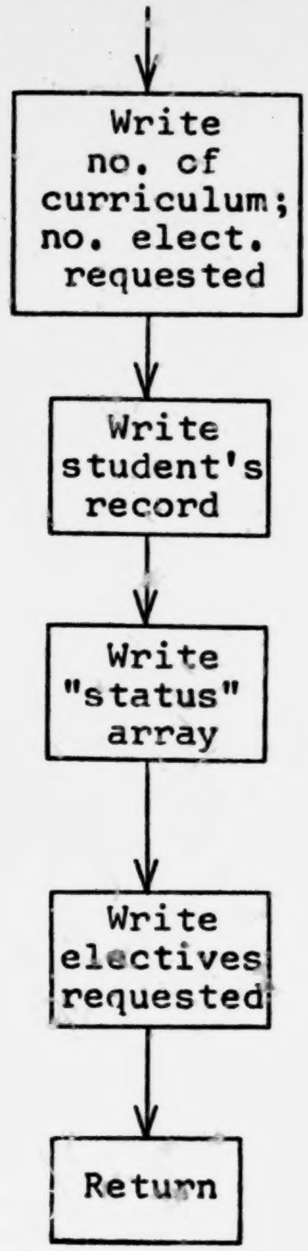

Read student name and number
Read a student record from master tape
Is it desired record?
Y
Calculate grades; revise record
Write Record on new master tape
N
Is it last record?
N
Write Record on new master tape
Y
Error
Return

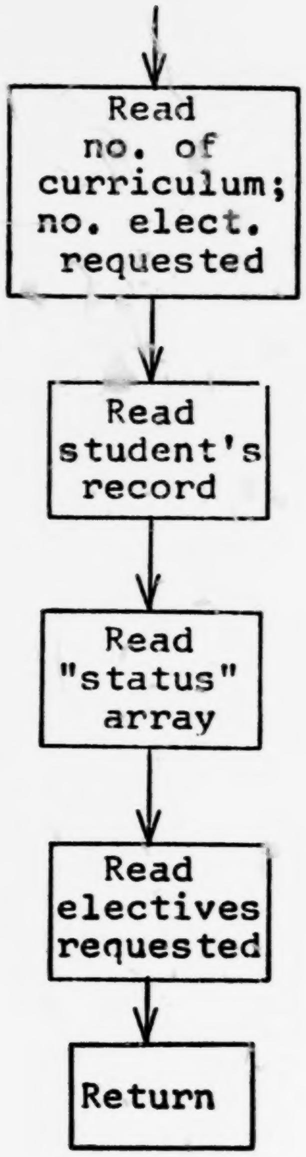
Figure 5

SUB-ROUTINE WRITE - (5)


Write no. of curriculum; no. elect. requested
Write student's record
Write "status" array
Write electives requested
Return

Writes one complete student record on tape

SUB-ROUTINE READ - (6)

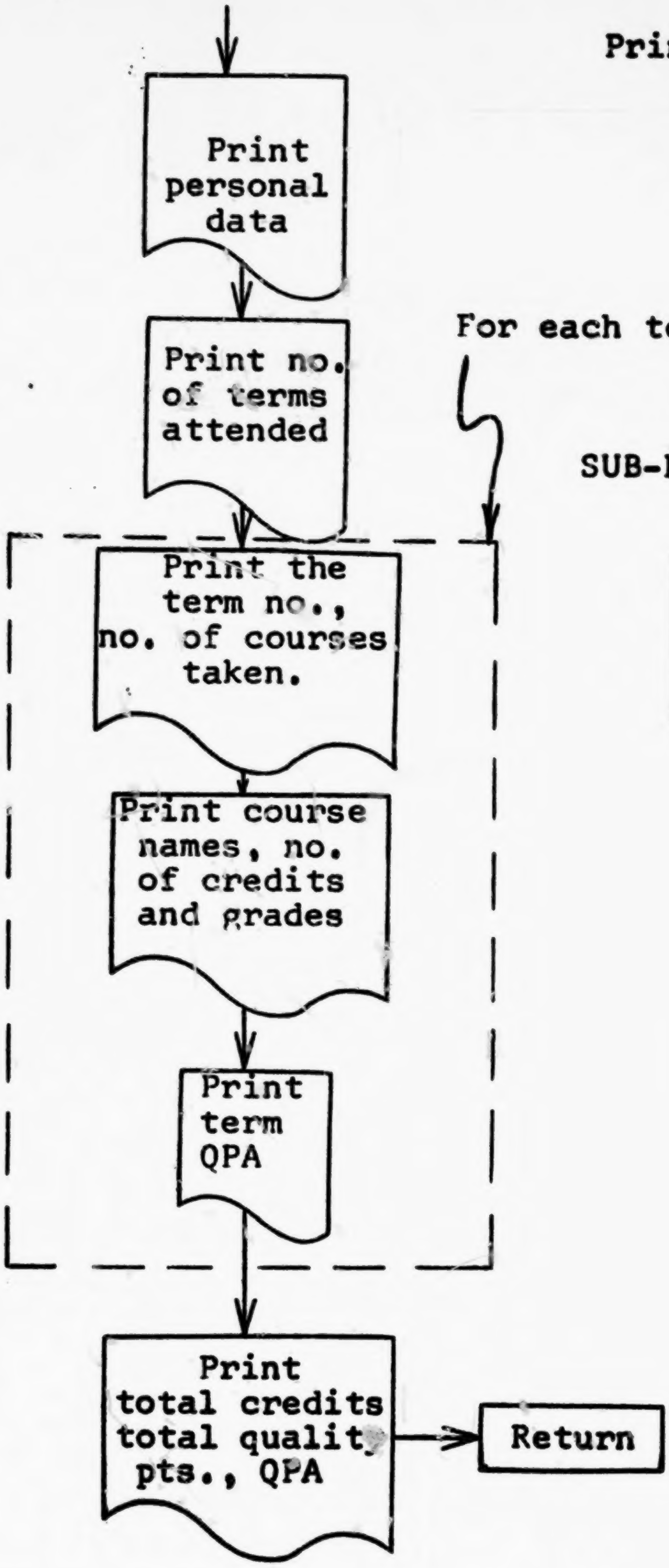

Read no. of curriculum; no. elect. requested
Read student's record
Read "status" array
Read electives requested
Return

Reads one complete student record from tape

Figure 6

SUB-ROUTINE PRINT - (7)

Prints the record of a given student.


Print personal data
Print no. of terms attended
For each term
Print the term no., no. of courses taken.
Print course names, no. of credits and grades
Print term QPA
Print total credits total quality pts., QPA
Return

SUB-ROUTINE PRREC - (8)

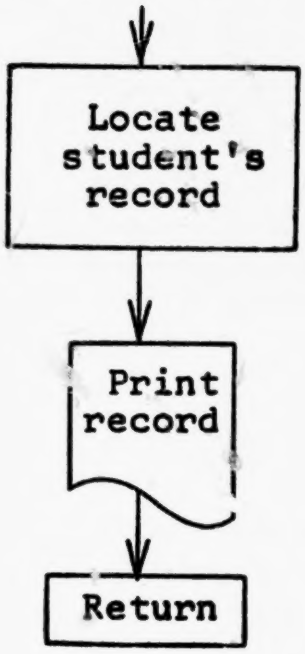

Locate student's record
Print record
Return

Figure 7

SUB-ROUTINE DMAND - (9)

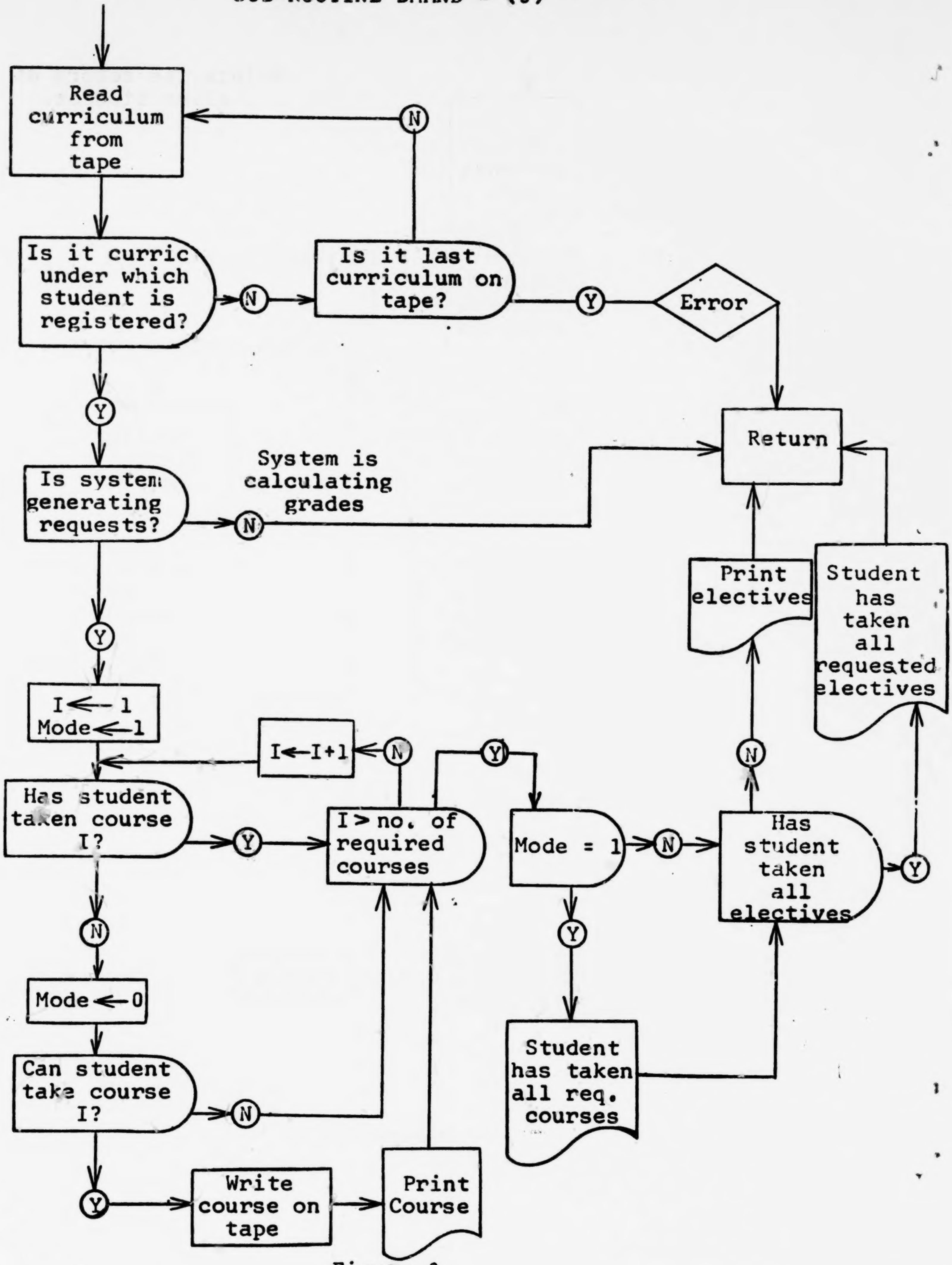

Read curriculum from tape
N
Is it curric under which student is registered?
N
Is it last curriculum on tape?
Y
Error
Y
Return
System is calculating grades
Is system generating requests?
N
Y
Print electives
Student has taken all requested electives
I ← 1
Mode ← 1
I ← I+1
N
Y
Has student taken course I?
Y
I > no. of required courses
Mode = 1
N
Has student taken all electives
Y
N
N
Y
Mode ← 0
Can student take course I?
N
Student has taken all req. courses
Y
Write course on tape
Print Course

Figure 8

SUB-ROUTINE GCALC - (10)

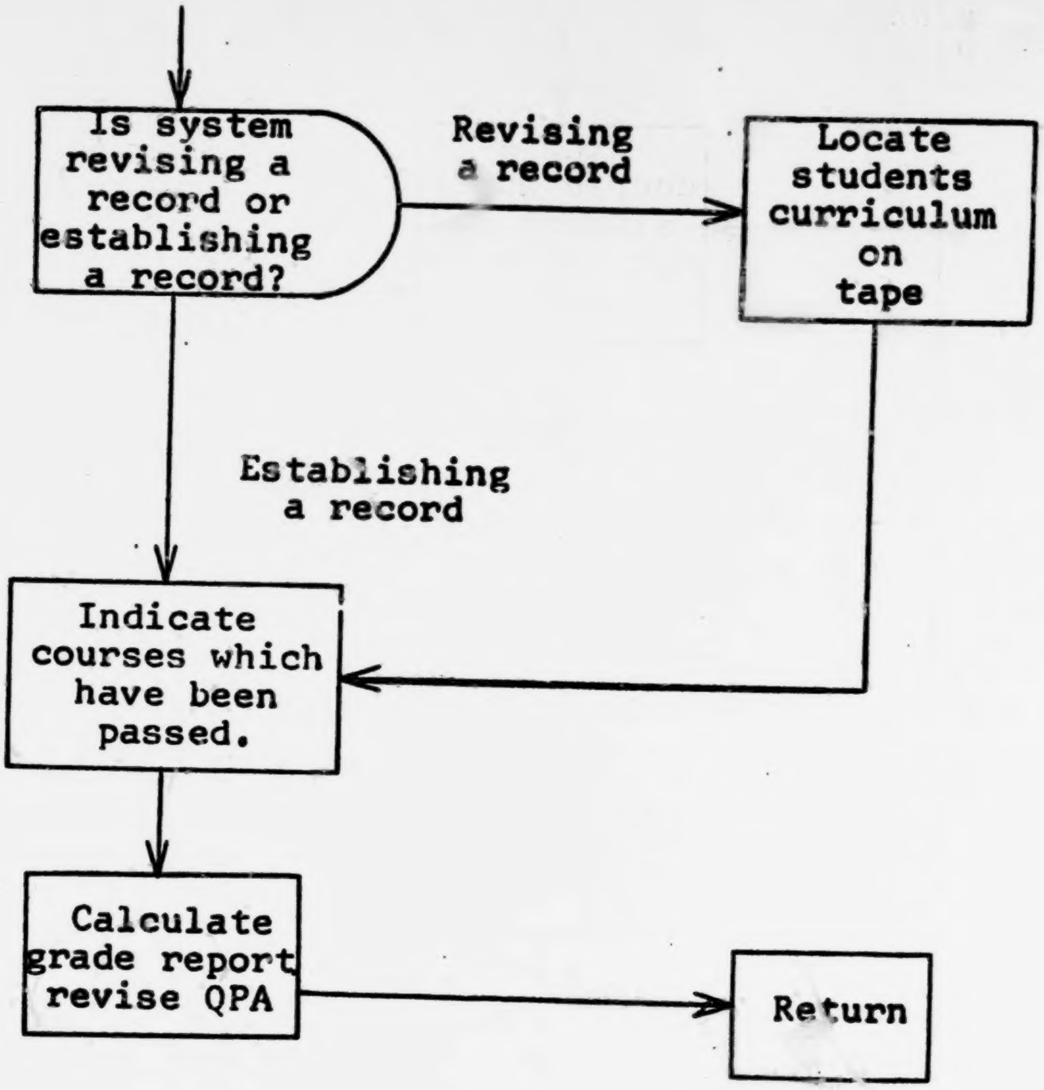

Is system revising a record or establishing a record?
Revising a record
Locate students curriculum on tape
Establishing a record
Indicate courses which have been passed.
Calculate grade report revise QPA
Return

Figure 9.

SUB-ROUTINE BINRY - (11)

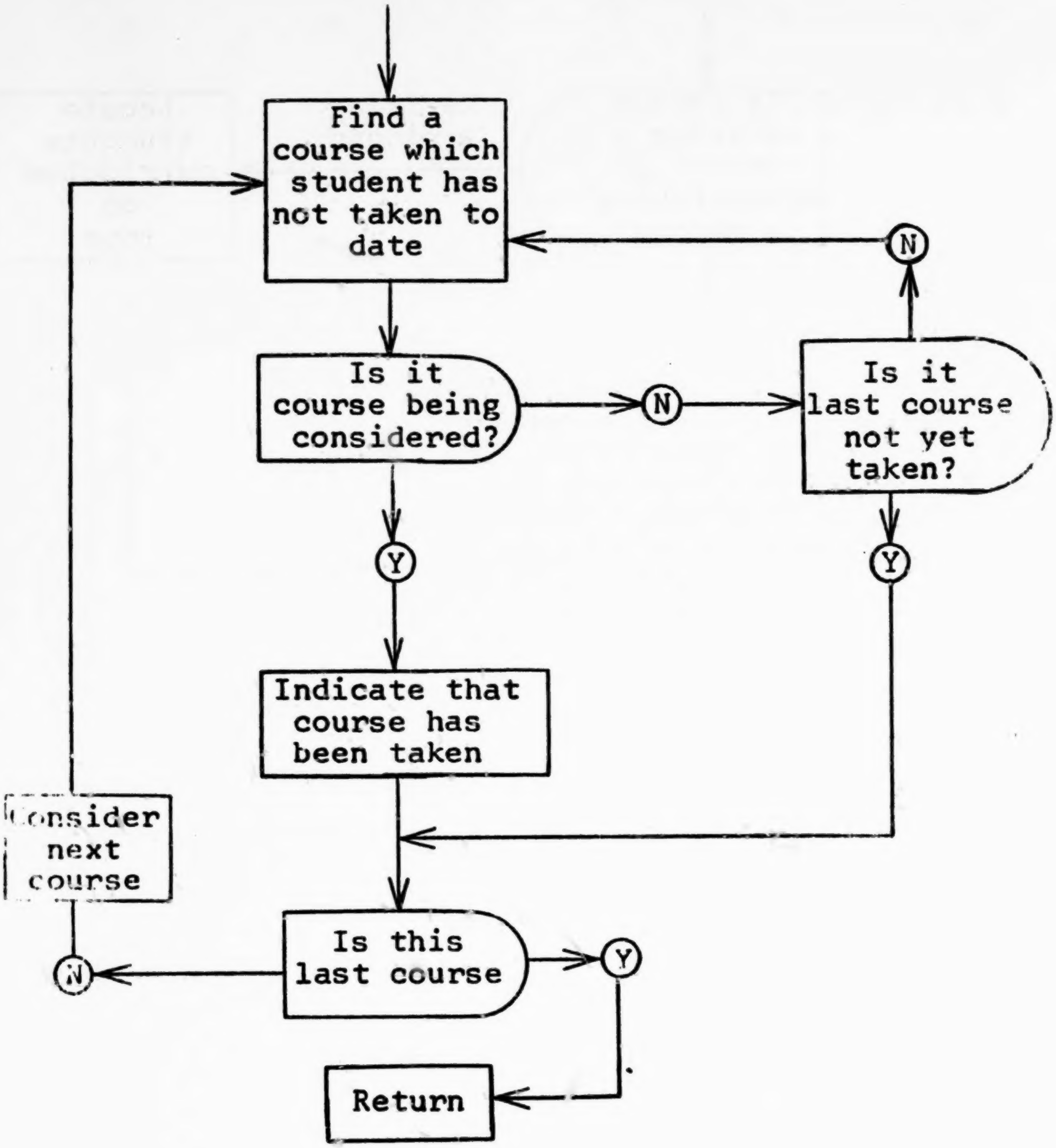

Find a course which student has not taken to date
Is it course being considered?
N
Is it last course not yet taken?
N
Y
Y
Indicate that course has been taken
Consider next course
Is this last course
N
Y
Return

Figure 10

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE

FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11

```
***  BAUGHMAN,WAYNE                                         ISUINDUE         590
*    COMPILE FORTRAN,EXECUTE FORTRAN,DUMP  IF ERROR

      SUBROUTINE ADDST
C     ADDS NEW STUDENTS RECORD TO PREVIOUSLY ESTABLISHED MASTER TAPE
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      DIMENSION IELEC(50),NCRDE(25),IA(1000),ISWCH(100),IEQCO(200)
      COMMON IELEC,NCRDE,NELEC,IA,IEQCO,ISWCH
      MONTR  = 1
      REWIND ITAP4
      REWIND ITAP3
      NOWDS=1
      READ 10,NOADD
   10 FORMAT (I5)
      READ TAPE ITAP4,NOOLD
      DO 20 I=1,NCOLD
      CALL READ (ITAP4,NOWDS)
   20 CALL WRITE (ITAP3,NOWDS)
      IMAST=ITAP3
      ITAP3=ITAP4
      ITAP4=IMAST
      DO 30 I=1,NOADD
   30 CALL ESTBL
      END FILE ITAP4
      RETURN
      END

      SUBROUTINE BINRY(JBEG,JEND)
C     INDICATES WHICH COURSES HAVE BEEN SUCESSFULLY COMPLETED BY THE STUDENT
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      MONTR  = 2
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      DIMENSIONIA(1000), ISWCH(100),IEQCO(200)
      COMMONIA,IEQCO,ISWCH,IOURS,IREQS
      EQUIVALENCE(N,M)
      NTEMP =NREQC*2
      NTEME= NELEC*2
      DO 25 I=JBEG,JEND,4
      M = 0
      DO 15 J=1,NTEMP,2
      M=M+1
    9 IF(ISWCH(M))990,10,15
   10 IF(IA(I)-IEQCO(J))15,11,15
   11 IF(IA(I+1)-IEQCO(J+1))15,12,15
   12 ISWCH(M)=1
      GO TO 25
   15 CONTINUE
  800 PRINT 850,IA(I),IA(I+1) ,IA(1),IA(2),IA(3),IA(4),NOCIR
      IF (NELEC)990,25,16
   16 DO 200J=1,NTEME,2
      N=N+1
  210 IF(ISWCH(N))990,215,200
  215 IF(IA(I)-IELEC(J))200,216,200
  216 IF(IA(I+1)-IELEC(J+1))200,12,200
  200 CONTINUE
  801 PRINT 851,IA(I),IA(I+1)  ,    (IA(J),J=1,4)
   25 CONTINUE
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE

FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
      RETURN
  850 FORMAT(2A5,37H COURSE IS NOT REQUIRED FOR STUDENT    ,4A5,
     125H WITH CURRICULUM NO.        ,I5)
  851 FORMAT(2A5,42H COURSE IS NOT AN ACCEPTABLE ELECTIVE FOR    ,4A5)
  990 STOP 990
      END

      SUBROUTINE CURIC
C     WRITES CURRICULUM INFORMATION ON TAPE
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      MONTR  = 3
      DIMENSION IEQCO(200),NCRED(100),IREQS(100),IOURS(100)
      COMMON IEQCO,       IREQS,IOURS,NUMCR,X,NDUMY,N2PRQ   ,NCRED
      READ 10, NOCIR , NREQC,NPREQ
      N2PRQ = 2*NPREQ
      NQC = NREQC*2
   10 FORMAT (8I10)
      IF(NOCIR-1)5,7,5
    5 READ TAPE ITP12,NUMCR
      READ TAPE ITP12,M
      READ TAPE ITP12,NDUMY
      IF(NUMCR -NDUMY)5,6,5
    6 BACKSPACE ITP12
    7 WRITE TAPE ITP12,NOCIR,NREQC,NPREQ
      J = -4
      DO 500 I =1,NQC,10
      J = J +5
  500 READ15,IEQCO(I),IEQCO(I+1),NCRED(J),IEQCO(I+2),IEQCO(I+3),NCRED(J+1)
     11),IEQCO(I+4),IEQCO(I+5),NCRED(J+2),IEQCO(I+6),IEQCO(I+7),NCRED(J+
     23),IEQCO(I+8),IEQCO(I+9),NCRED(J+4)
   15 FORMAT(2A5,I5,2A5,I5,2A5,I5,2A5,I5,2A5,I5)
      READ 20, (IOURS(I),IOURS(I+1),IREQS(I),IREQS(I+1),I=1,N2PRQ,2)
   16 WRITE TAPE ITP12,(IEQCO(I),I=1,NQC),(NCRED(I) ,I=1,NREQC),
     1(IOURS(I),I=1,N2PRQ),(IREQS(I),I=1,N2PRQ)
   20 FORMAT(16A5)
      WRITE TAPE ITP12,NOCIR
      END FILE ITP12
      REWIND ITP12
      RETURN
      END

      SUBROUTINE CMAND
C     GENERATES COURSE REQUESTS
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      MONTR  = 4
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      DIMENSIONIA(1000),IEQCO(200),NCRED(100),ISWCH(100),IOURS(200),
     1IREQS(200)
      COMMONIA,IEQCO,ISWCH,IOURS,IREQS
    5 READ TAPE ITP12,NUMCR,NREQC,NPREQ
      IF(NUMCR-NOCIR)6,10,6
    6 N2PRQ =NPREQ* 2
      NQC = NREQC*2
      READ TAPE ITP12,(IEQCO(I),I=1,NQC),(NCRED(I),I=1,NREQC),
     1(IOURS(I),I=1,N2PRQ),(IREQS(I),I=1,N2PRQ)
    8 READ TAPE ITP12,M
      IF(M-NUMCR)9,7,9
    9 IF(M-NOCIR)13,12,13
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE

FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
  12 BACKSPACE ITP12
     GO TO 5
  13 NUMCR=M
     GO TO 6
   7 PRINT 69,NOCIR
  69 FORMAT(16H CURRICULUM NO. ,I5,15H IS NOT ON TAPE //)
     STOP 68
  10 N2PRQ =NPREQ* 2
     NQC = NREQC*2
     READ TAPE ITP12,(IEQCO(I),I=1,NQC),(NCRED(I),I=1,NREQC),
    1(IOURS(I),I=1,N2PRQ),(IREQS(I),I=1,N2PRQ)
     IF(SENSE LIGHT 3 )60,11
  11 MCDE = 1
     N = 0
     IZK = 1
     DO 50 I=1,NCC   ,2
     N=N+1
     IF(ISWCH(N))990,15,50
  15 MODE = 0
     DO 45 K = 1,N2PRQ,2
     IF(IEQCO(I) -IOURS(K))45,20,45
  20 IF(IEQCO(I+1)-ICURS([illegible]+1))45,21,45
  21 M=0
     DO 30 J=1,NCC   ,2
     M=M+1
     IF(IEQCO(J)-IREQS(K))30,25,30
  25 IF(IEQCO(J+1)-IREQS(K+1))30,26,30
  26 IF(ISWCH(M))990,50,45
  30 CONTINUE
  45 CONTINUE
     PRINT 800,IA(1),IA(2),IA(3),IA(4),IEQCO(I),IEQCO(I+1),NOCIR
 800 FORMAT(4A5,18H CAN TAKE COURSE  ,2A5,20HHIS CURRICULUM NO IS,  I5)
     IF (IZK-NDESC)46,60,60
  46 IZK = IZK+1
  50 CONTINUE
     IF(MODE)990,59,80
  80 PRINT 90
  90 FORMAT(46H THE STUDENT HAS TAKEN ALL REQUIRED COURSES          //)
  59 MODE=1
     IF(NELEC) 990,60,62
  62 NQC = NELEC*2
     DO 250 I=1,NQC,2
     N=N+1
     IF(ISWCH(N))990,215,250
 215 MCDE=0
     PRINT 1000,(IA(J),J=1,4),IELEC(I),IELEC(I+1)
     IF (IZK-NDESC) 245,60,60
 245 IZK = IZK+1
 250 CONTINUE
     IF(MODE)990,60,81
  81 PRINT 1001,(IA(J),J=1,4)
1001 FORMAT(4A5,39H HAS TAKEN ALL REQUESTED ELECTIVES      //)
1000 FORMAT(4A5,20H CAN TAKE ELECTIVE  ,2A5/)
  60 REWIND ITP12
     RETURN
 990 STOP 990
     END

     SUBROUTINE ESTBL
     ESTABLISHES THE STUDENT RECORDS ON TAPE WHEN THE SYSTEM IS INSTALLED
     COMMON NDESC
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE

FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      DIMENSION IELEC(50),NCRDE(25),IA(1000),ISWCH(100),IEQCO(200)
      COMMON IELEC,NCRDE,NELEC,IA,IEQCO,ISWCH
   15 FORMAT (16I5)
   16 FORMAT (16A5)
      MONTR  = 5
      READ 16,(IA(K),K=1,48)
      READ 15,NUMTM,NOCIR,NELEC,NDESC
      IF (NELEC)991,18,14
   14 L = 0
      K=NELEC*2
      DO 25 J=1,K,2
      L=L+1
   25 READ 17, IELEC(J),IELEC(J+1).NCRDE(L)
   17 FORMAT(2A5,I5)
   18 DO 60 K =1,100
   60 ISWCH(K) = 0
      SENSE LIGHT 3
      CALL DMAND
      J=49
      IA(J)=NUMTM
      TCRED=0.0
      TQPTS=0.0
      IF(NUMTM)991,48,49
   48 J=J+1
      QPA=0.0
      GO TO 51
   49 DO 50 NTERM = 1,NUMTM
      SENSE LIGHT 4
      READ 15,NOCRS
      IA(J+1)=NOCRS
      J=J+2
      NUREV = 4*NOCRS
      NOLIM = NUREV + J  -1
      READ 16,(IA(L), L= J, NOLIM)
   50 CALL GCALC (J, NOCRS , TCRED,TQPTS,NOLIM)
      J = J+1
      QPA=TQPTS/TCRED
   51 IA(J)=TQPTS
      IA(J+1)=TCRED
      IA(J+2)=QPA *10000.0
      J=J+2
      CALL PRINT
  100 CALL WRITE (ITAP4,J) .
      RETURN
  991 STOP 991
      END

      SUBROUTINE GCALC(J,NOCRS,TCRED,TQPTS,NOLIM)
C     CALCULATES GRADES
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      MONTR  = 6
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      DIMENSIONIA(1000),ISWCH(100),IEQCO(200)
      COMMONIA,IEQCO,ISWCH,IOURS,IREQS
      SUBROUTINE CALCULATES TOTAL CREDITS, TOTAL QUALITY POINTS, AND
      INTERNALLY CALCULATES AND RECORDS QPA FOR THE TERM.
      PARAMETERS OF SUBROUTINE ARE J=THE LOCATION OF FIRST COURSE IN
      THE A(I) VECTOR WHICH IS TO BE EVALUATED.  NOCRS IS THE NUMBER OF
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
C     COURSES TO BE EVALUATED. NOLIM IS THE LOCATION OF THE GRADE FOR
C     THE LAST COURSE THAT IS TO BE EVALUATED = 4 X NOCRS + J.
C     THE VALUE OF J THAT IS RETURNED BY THE SUBROUTINE IS THE FIRST
C     UNASSIGNED LOCATION IN THE A(I) VECTOR. IN MOST CASES TOTAL NO. OF
C     QUALITY POINTS WILL BE IN THIS LOCATION.
      NOQPS=0
      NCRED=0
      IF(SENSE LIGHT 4 ) 10,9
    9 SENSE LIGHT 3
      CALL DMAND
   10 JJJ= J-2
      CALL BINRY(J,NOLIM)
      JJJ=J-2
      DO 28 M=1,NOCRS
      JJJ=JJJ+4
      N = IA(JJJ)-90
      J = JJJ+1
      IGOTO =(IA(J)-6000)/100
      GO TO(20,21,22,23,24,25),IGOTO
   20 NOQPS=NOQPS+N*4
      GO TO 28
   21 NOQPS=NOQPS+N*3
      GO TO 28
   22 NOQPS=NOQPS+N*2
      GO TO 28
   23 NOQPS=NOQPS+N*1
      GO TO 28
   24 STOP
   25 NTEMP=NREQC*2
      MTEZ = 0
      DO 100 IZ=1,NTEMP,2
      MTEZ=MTEZ+1
      IF(IEQCO(IZ)-IA(JJJ-2))100,30,100
   30 IF(IEQCO(IZ+1)-IA(JJJ-1))100,31,100
   31 ISWCH(MTEZ)=0
      GO TO 28
  100 CONTINUE
      NTEMP = NELEC*2
      DO 110 IZ = 1,NTEMP,2
      MTEZ = MTEZ+1
      IF (IELEC(IZ)-IA(JJJ-2))110,101,110
  101 IF(IELEC(IZ+1)-IA(JJJ-1))110,102,110
  102 ISWCH(MTEZ) =0
      GO TO 28
  110 CONTINUE
      PRINT 115,IA(JJJ-2),IA(JJJ-1)
  115 FORMAT(2A5,40HCOURSE IS NOT REQUIRED ,OUTSTANDING F     ///)
   28 NCRED=NCRED+N
      TMQPA=FLOTF(NOQPS)/FLOTF(NCRED)*10000.0
      J=NOLIM+1
      IA(J)   =TMQPA
      TCRED=TCRED+FLOTF(NCRED)*10.0
   50 TQPTS=TQPTS+FLOTF(NOQPS)*10.0
      RETURN
      END

      SUBROUTINE PRINT
C     PRINTS ONE STUDENTS RECORD
      COMMON NDESC
C     SUBROUTINE PRINTS ONE STUDENT RECORD FROM MEMORY.
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE
FILE MAINTENANCE FOR STUDENT RECORDS
Figure 11 (cont.)

```
      MUNTR  = 7
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC,NCRDE,NELEC
   11 FORMAT(25H QUALITY POINT AVERAGE IS  , F6.2)
   12 FORMAT(30H THIS STUDENT ATTENDED SCHOOL  ,I3, 6H TERMS)
   13 FORMAT ( 12H TERM NUMBER,I4,  1H,, I4 ,52H COURSES WERE TAKEN.
     1THEY ARE AS FOLLOWS...       // )
   16 FORMAT (16A5)
   17 FORMAT ( 14H GPA FOR TERM  ,I3,4H IS ,F6.2)
   18 FORMAT(17H TOTAL CREDITS =  ,F6.2)
   19 FORMAT( 23H TOTAL QUALITY POINTS =  ,F6.2)
      DIMENSIONIA(1000)
      COMMONIA
      PRINT 16,(IA(I),I=1,48)
      NUMTM=IA(49)
      PRINT 12,NUMTM
      J=48
      DO 150 II=1,NUMTM
      J=J+2
      NOCRS=IA(J)
      PRINT 13,II    ,NOCRS
      DO 130 I=1,NOCRS
      PRINT 16,IA(J+1),IA(J+2),IA(J+3),IA(J+4)
  130 J=J+4
      ZZZ = FLOATF(IA(J+1))/10000.0
      PRINT 17,II,ZZZ
  150 CONTINUE
      ZZZ = FLOATF(IA(J+4))/10000.0
      PRINT 11,ZZZ
      ZZZ = FLOATF(IA(J+3))/10.
      PRINT 18,ZZZ
      ZZZ = FLOATF(IA(J+2))/10.00
      PRINT 19,ZZZ
      PRINT 20
   20 FORMAT (C 2)
      RETURN
      END

      SUBROUTINE PRREC
C     LOCATES AND PRINTS A PARTICULAR STUDENTS RECORD
      COMMON NOESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MUNTR
      DIMENSION IELEC(50),NCRDE(25),IA(1000),ISWCH(100),IEQCO(200)
      COMMON IELEC,NCRDE,NELEC,IA,IEQCO,ISWCH
      MUNTR = 11
      REWIND ITAP4
      READ TAPE ITAP4,NOSTU
      READ 10,N1,N2,N3,N4,NO1,NO2
   10 FORMAT(16A5)
      DO 20 I=1,NOSTU
      CALL READ (ITAP4,NOWDS)
      IF(IA(5)-NO1)20,11,20
   11 IF(IA(6)-NO2)20,12,20
   12 CALL PRINT
      REWIND ITAP4
      RETURN
   20 CONTINUE
      PRINT 60,N1,N2,N3,N4,NO1,NO2
   60 FORMAT (4A5,18H WITH STUDENT NO. ,2A5,69H IS NOT ON THE MASTER TAP
     1E OR THE STUDENT NO. IS PUNCHED IN THE WRONG /20X,27H COLUMN ON TH
     2E INPUT DATA       /)
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
   61 REWIND ITAP4
      RETURN
      END

      SUBROUTINE READ (K,NOWDS)
C     READS ONE STUDENTS RECORD FROM MASTER TAPE
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      MONTR  = 8
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      DIMENSIONIA(1000),ISWCH(300)
      COMMONIA,ISWCH
C     SUBROUTINE READS ONE STUDENT RECORD FROM MASTER TAPE.
      READ TAPE K,NOWDS,NOCIR,NREQC,NDESC,(IA(I),I=1,NOWDS),N,(ISWCH(I),
     1I=1,N),NLC2,(IELEC(I),I=1,NLC2),NELEC,(NCRDE(I),I=1,NELEC)
  500 RETURN
      END

      SUBROUTINE REVIS
C     BRINGS MASTER FILE UP TO DATE
      COMMON NDESC
      COMMON NOCIR,NREQC,NPREQ,ITAP3,ITAP4,ITP12,ITP14,MONTR
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      MONTR=9
      DIMENSIONIB(500),IA(1000)
      DIMENSION ISWCH(100),IEQCO(200)
      COMMONIA,IEQCO,ISWCH
C     SUBROUTINE WILL UPDATE STUDENTS RECORD. THE INPUT IS THE STUDENT
C     NUMBER, THE NUMBER OF COURSES TAKEN AND THE COURSE INFORMATION.
   10 FORMAT(16A5)
   11 FORMAT(2A5,I5)
   15 FORMAT(16I5)
   16 FORMAT (8F10.2)
   75 FORMAT (90H THIS STUDENTS RECORD IS NOT ON TAPE OR STUDENT NUMBER
     1IS PUNCHED IN THE WRONG COLUMNS        //)
      READ 10,(IB(L),L=1,6 )
      READ TAPE ITAP4,NOSTU
      NOCNT=0
      WRITE TAPE ITAP3,NOSTU
   19 CALL READ (ITAP4,NOWDS)
      NOCNT=NOCNT+1
      IF(NOSTU-NOCNT)50,21,21
C     SEARCH FOR STUDENTS RECORD.
C     A(5) AND A(6) ARE THE STUDENT NUMBERS ON THE MASTER TAPE.
C     B(5) AND B(6) ARE THE PARTICULAR STUDENTS NUMBER.
   21 IF(IA(5)-IB(5))20,25,20
   25 IF(IA(6)-IB(6))20,30,20
   20 CALL WRITE (ITAP3,NOWDS)
      GO TO 19
   50 PRINT 75
      STOP 50
   30 IF(SENSE LIGHT 1)31,35
   31 READ 15,NELEC
      KKK  = NELEC *2
      L=0
      DO 27 J=1,KKK,2
      L=L+1
   27 READ 11,IELEC(J),IELEC(J+1),NCRDE(L)
      GO TO 199
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE
FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
   35 READ 15,NOCRS
      IB( 7)=NOCRS
      J = 9
      NOREV=4*NOCRS
      NOLIM=NOREV+J-1
      READ 10,(IB(L),L=J,NOLIM)
      NTHST=NOCNT
      GPA=IA(NOWDS)
      TCRED=IA(NOWDS-1)
      TQPTS=IA(NOWDS-2)
      IA(49)=IA(49)+1
      K=NOWDS-2
C     KMAX IS THE NUMBER OF WORDS THAT WILL BE ON STUDENTS RECORD
      KMAX=K+NOREV
C     A(K) IS THE NUMBER OF COURSES TAKEN LAST TERM.
      IA(K)=NOCRS
      K=K+1
      KK=K-8
C     REVISES A(I) VECTOR OF THE STUDENT.
      DO 200 J=K,KMAX
      KKK = J-KK
  200 IA(J)=IB(KKK)
      CALL GCALC(K,NOCRS,TCRED,TQPTS,KMAX)
      IA(KMAX+2)=TQPTS
      IA(KMAX+3)=TCRED
      IA(KMAX+4)=TQPTS/TCRED *10000.0
      NOWDS=KMAX+4
C     PLACE UPDATED RECORD ON MASTER TAPE
  199 NOCNT=NOCNT+1
      CALL WRITE(ITAP3,NOWDS)
      CALL PRINT
      IF (NOCNT-NCSTU) 202,202,203
  202 DO 201 J=NOCNT,NOSTU
      CALLREAD(ITAP4,NOWDS)
  201 CALL WRITE(ITAP3,NOWDS)
  203 END FILE ITAP3
      IMAST=ITAP3
      ITAP3=ITAP4
      ITAP4=IMAST
  330 REWIND ITAP4
      REWIND ITAP3
      RETURN
      END

      SUBROUTINE WRITE (K,NOWDS)
C     WRITES ONE STUDENTS RECORD ONTO MASTER TAPE
      COMMON NOESC
      COMMON NOCIR,NREQC,NPREG,ITAP3,ITAP4,ITP12,ITP14
      COMMON MONTR
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      DIMENSIONIA(1000),ISWCH(300)
      COMMONIA,ISWCH
      MONTR=10
      NREGC = NREGC +NELEC
      NLC2 = NELEC*2
      WRITETAPE K,NOWDS,NOCIR,NREGC,NOESC,(IA(I),I=1,NOWDS),NREOC,(ISWCH
     1(I),I=1,NREGC),NLC2,(IELEC(I),I=1,NLC2),NELEC,(NCRDE(I),I=1,NELEC)
      RETURN
      END
```

PROGRAM TO GENERATE COURSE REQUESTS AND PROVIDE

FILE MAINTENANCE FOR STUDENT RECORDS

Figure 11 (cont.)

```
** MAIN PROGRAM

C     READS INSTRUCTIONS, DETERMINES CALLING SEQUENCE FOR SUBROUTINES
      COMMON NDESC
      COMMON NOCIR,NREGC,NPREC,ITAP3,ITAP4,ITP12,ITP14,MOJTR
      MOJTR=11
      DIMENSION IELEC(50),NCRDE(25)
      COMMON IELEC, NCRDE,NELEC
      ITAP3=3
      ITAP4=4
      ITP12=12
      REWIND ITP12
      REWIND ITAP4
      DIMENSIONIA(1000),ISWCH(100),IEQCU(200)
      COMMONIA ,IEQCU,ISWCH
   14 FORMAT(//)
   15 FORMAT(16I5)
   16 FORMAT(16A5)
   17 FORMAT(C 2)
      SENSE LIGHT 0
  295 READ 300,NCODE,NUMDC,INDIC
  300 FORMAT(A5,    2I5)
      IF(NUMDC)995,310,320
  310 NUMDC=1
  320 IF(NCODE-79)325,365,325
  325 IF(NCODE-64)326,375,326
  326 IF(NCODE-6365)330,430,330
  330 IF(NCODE-77)335,385,335
  335 IF(NCODE-61)340,395,340
  340 IF(NCODE-63)345,405,345
  345 IF(NCODE-65)350,415,350
  350 IF(NCODE-82)355,425,355
  355 PRINT 360,NCODE
  360 FORMAT(A5,31H IS AN INVALID INSTRUCTION CODE   //)
      GO TO 295
  365 DO 370 I=1,NUMDC
  370 CALL REVIS
      GO TO 295
  375 READ TAPE ITAP4,NOSTU
      DO 380 I=1,NOSTU
      CALL READ  (ITAP4,NCWDS)
  380 CALL DMAND
      PRINT 17
      GO TO 295
  385 DO 390 I=1,NUMDC
  390  CALL PRREC
      GO TO 295
  395 DO 400 I=1,NUMDC
  400 CALL AUDST
      GO TO 295
  405 DO 410 I=1,NUMDC
  410 CALL CURIC
      GO TO 295
  415 NOSTU=NUMDC
      WRITE TAPE ITAP4,NOSTU
      DO 420 I=1,NOSTU
  420 CALL ESTBL
      END FILE ITAP4
      REWIND ITAP4
      GO TO 295
```

```
425 STOP
430 DO 435 I=1,NUMDC
    SENSE LIGHT 1
435 CALL REVIS
    GO TO 295
995 STOP 995
    END

  C    2


                         CURRICULUM NO. 1 COURSES

          1          34          19
E G 1            2MATH 22    4E G 2        2I E 22        3MATH 23       4
PHYS 95          4I E 121    3I E 33       3MATH 24       4PHYS 96       4
T A M 23         3M E 23     3I E 171      3MATH 110      3PHYS 97       4
T A M 24         3I E 172    3T A M 27     3PSYCH 132     2I E 50        3
I E 42           3I E 122    3I E 178      3I E 51        3I E 180       3
E E 11           3I E 52     3I E 97       1M E 50        3I E 43        3
I E 96           1E E 12     3I E 23       3MET E 9       3

COURSE    PREREQ.   COURSE    PREREQ.   COURSE    PREREQ.   COURSE    PREREQ.

MATH 23   MATH 22   I E 121   MATH 23   MATH 24   MATH 23   PHYS 96   PHYS 95
M E 23    PHYS 95   I E 171   MATH 24   MATH 110  MATH 24   PHYS 97   PHYS 96
T A M 24  T A M 23  I E 172   I E 171   T A M 27  PHYS 95   I E 122   I E 121
I E 175   I E 171   I E 180   I E 179   E E 11    PHYS 97   I E 97    I E 171
M E 50    PHYS 97   I E 98    I E 97    E E 12    E E 11

                         CURRICULUM NO. 2 COURSES

         2          42          33
ENG 1            3MATH 21    4E G 1        2CHEM 11       4ECON 83       3
P ED 1           1ENG 2      3MATH 22      4E G 2         2CHEM 12       4
P ED 2           1I E 22     3MATH 23      4PHYS 95       4I E 121       3
I E 33           3MATH 24    4PHYS 96      4T A M 23      3M E 23        3
I E 171          3MATH 110   3PHYS 97      4T A M 24      3I E 172       3
ACCT 77          3T A M 27   3PSYCH 132    2I E 50        3I E 42        3
I E 122          3I E 178    3I E 51       3I E 180       3E E 11        3
I E 52           3I E 97     1ENG 192      2M E 50        3I E 43        3
I E 96           1E E 12    .3

COURSE    PREREQ.   COURSE    PREREQ.   COURSE    PREREQ.   COURSE    PREREQ.

ENG 2     ENG 1     MATH 22   MATH 21   E G 2     E G 1     CHEM 12   CHEM 11
P ED 1    P ED 2    I E 22    MATH 21   MATH 23   MATH 22   PHYS 95   MATH 22
I E 121   MATH 22   MATH 24   MATH 23   PHYS 96   PHYS 95   T A M 23  PHYS 95
M E 23    PHYS 95   I E 171   MATH 23   MATH 110  MATH 24   PHYS 97   PHYS 96
T A M 24  T A M 23  T A M 24  PHYS 96   I E 172   I E 171   T A M 27  T A M 23
I E 50    I E 33    I E 42    ACCT 77   I E 122   I E 121   I E 178   I E 171
I E 50    I E 51    I E 180   I E 179   E E 11    PHYS 97   I E 52    I E 51
I E 97    I E 178   M E 50    PHYS 97   I E 43    I E 42    I E 98    I E 97
E E 12    E E 11
    E    4
CRAWFORD, RONALD E. E-078-471     70 EASY STREET
GE 7-2103   UNIONTOWN SENIOR HIGH SCHOOL
DECEMBER 8, 1939
```

SAMPLE INPUT DATA

Figure 11 (cont.)

SAMPLE INPUT DATA

Figure 11 (cont.)

```
    7    2    5    6
** ELECTIVES  REQUESTED
POL SC 1      3
POL SC 2      2
CH E 83       4
INTDIS 81     3
PHIL 1        3
    5
** COURSES TAKEN
CHEM 11       4   C E G 1          2   C ENG 1          3   B MATH 21        4   B
POL SC 1      3   C
    5
ENG 2         3   B E G 2          2   B MATH 22        4   A CHEM 12        4   C
POL SC 2      3   C
    4
ECON 83       3   B PHYS 95        4   F P ED 1         1   A MATH 23        4   A
    3
MATH 24       4   A PHYS 96        4   B CH E 83        4   B
    4
T A M 23      3   B MATH 110       3   A PHYS 95        4   A INTDIS 81      3   B
    4
PHYS 97       4   C M E 50         3   C ENG 192        2   A T A M 27       3   B
    4
I E 33        3   B ACCT 77        3   A T A M 24       3   D I E 171        3   C
BIONDI,RICHARD F. JR3E-030-524
BOX 356 FREDERICKTON,PA
ATTENDED BETHLEHEM CENTER
    7    2    5    4
** ELECTIVES  REQUESTED
M E 22
POL SC 1      3
POL SC 2      3
PHIL 1        3
M E 40        1
    5
** COURSES TAKEN
CHEM 11       4   B ENG 1          3   B MATH 21        4   B POL SC 1       3   C
E G 1         2   C
    5
E G 2         2   B MATH 22        4   B CHEM 12        4   A ENG 2          3   B
POL SC 2      3   C
    5
PHIL 1        3   B M E 22         3   D PHYS 95        4   C M E 40         1   B
MATH 23       4   B
    5
P ED 1        1   B MATH 24        4   B M E 50         3   D T A M 23       3   D
PHYS 96       4   C
    2
T A M 24      3   B MATH 110       3   A
    6
ECON 83       3   B I E 33         3   B I E 22         3   B P ED 2         1   A
PHYS 97       4   D T A M 27       3   B
    6
ACCT 77       3   B I E 121        3   A I E 171        3   B ENG 192        2   B
PSYCH 132     2   B E C 71         3   F
WILKINSON, ROBERT E. F-424-489     21 EDGEWOOD DRIVE, NORWALK,OHIO
MALE    DECEMBER 27,1936  SINGLE
ATTENDED OHIO WESLEYAN
    3    1    3    5
** ELECTIVES  REQUESTED
G B A 252     3
```

```
PHIL 7         3
ENG 21         3
    4
** COURSES TAKEN
I E 43         3   B PLT E 7        3   B MATH 22       4   B E G 1         2   B
    4
PHYS 35        4   C MATH 23        4   B I E 22        3   A I E 23        4   A
    5
MATH 24        4   A I E 171        3   B F A M 23      3   B PHYS 95       4   C
PSYCH 132      2   C
LUTHER,RICHARD D.     G-240-325
111 LAFAYETTE AVE.
PITTSBURGH 13, PA.
    1    2    4    5
** ELECTIVES  REQUESTED
ECON 80        3
PHIL 4         3
PSYCH 15       2
IND 80         3
    5
** COURSES TAKEN
ENG 1          3   C CHEM 11        4   C MATH 21       4   A E G 1         2   B
ECON 90        3   C
    A    4
WILKINSON, ROBERT E. F-424-487
** COURSES TAKEN LAST TERM
    5
I E 172        3   A I E 178        3   B I E 121       3   C MATH 110      3   A
E G 2          2   B
LUTHER,RICHARD D.     G-240-325
** COURSES TAKEN LAST TERM
    4
CHEM 12        4   C ENG 2          3   C E G 2         2   B MATH 22       4   A
HIGAUT,RICHARD F. JR3C-330-526
** COURSES TAKEN LAST TERM
    5
I E 178        3   C I E 172        3   A I E 51        3   B I E 1[illegible]2      3   A
E E 11         3   B
CRAWFORD, DONALD E. E-378-471
** COURSES TAKEN LAST TERM
    5
PHIL 2         1   A I E 172        3   B PHIL 1        3   A I E 22        3   B
I E 42         3   B
    B    4
    5
```

SAMPLE INPUT DATA

Figure 11 (cont.)

```
POL SC 1    COURSE IS NOT REQUIRED FOR STUDENT  CRAWFORD, RONALD E.  WITH CURRICULUM NO.        2
POL SC 2    COURSE IS NOT REQUIRED FOR STUDENT  CRAWFORD, RONALD E.  WITH CURRICULUM NO.        2
CH E 83     COURSE IS NOT REQUIRED FOR STUDENT  CRAWFORD, RONALD E.  WITH CURRICULUM NO.        2
INTDIS 81   COURSE IS NOT REQUIRED FOR STUDENT  CRAWFORD, RONALD E.  WITH CURRICULUM NO.        2
CRAWFORD, RONALD E. E-078-471      70 EASY STREET
GE 7-2103   UNIONTOWN SENIOR HIGH SCHOOL
DECEMBER 8, 1939
 THIS STUDENT ATTENDED SCHOOL    7 TERMS
 TERM NUMBER   1,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


CHEM 11         4   C
E G 1           2   C
ENG 1           3   B
MATH 21         4   B
POL SC 1        3   C
 GPA FOR TERM    1 IS   2.44
 TERM NUMBER   2,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ENG 2           3   B
E G 2           2   B
MATH 22         4   A
CHEM 12         4   C
POL SC 2        3   C
 GPA FOR TERM    2 IS   2.81
 TERM NUMBER   3,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ECON 80         3   B
PHYS 95         4   F
P ED 1          1   A
MATH 23         4   A
 GPA FOR TERM    3 IS   2.42
 TERM NUMBER   4,   3 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


MATH 24         4   A
PHYS 96         4   B
CH E 83         4   B
 GPA FOR TERM    4 IS   3.33
 TERM NUMBER   5,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


T A M 23        3   B
MATH 110        3   A
PHYS 95         4   A
INTDIS 81       3   B
 GPA FOR TERM    5 IS   3.54
 TERM NUMBER   6,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


PHYS 97         4   C
M E 50          3   C
ENG 192         2   A
T A M 27        3   B
 GPA FOR TERM    6 IS   2.58
 TERM NUMBER   7,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I E 33          3   B
ACCT 77         3   A
```

SAMPLE OUTPUT

Figure 12

```
[illegible]          3   D
I = 1/1              3   C
 QPA FOR TERM    7 IS    2.50
 QUALITY POINT AVERAGE IS   2.80
 TOTAL CREDITS =   93.00
 TOTAL QUALITY POINTS =260.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
POL SC 1    COURSE IS NOT REQUIRED FOR STUDENT  BIONDI,RICHARD F. JR WITH CURRICULUM NO.        2
POL SC 2    COURSE IS NOT REQUIRED FOR STUDENT  BIONDI,RICHARD F. JR WITH CURRICULUM NO.        2
PHIL 1      COURSE IS NOT REQUIRED FOR STUDENT  BIONDI,RICHARD F. JR WITH CURRICULUM NO.        2
M E 22      COURSE IS NOT REQUIRED FOR STUDENT  BIONDI,RICHARD F. JR WITH CURRICULUM NO.        2
M E 40      COURSE IS NOT REQUIRED FOR STUDENT  BIONDI,RICHARD F. JR WITH CURRICULUM NO.        2
BIONDI,RICHARD F. JR3E-030-524
BOX 356  FRED ERICKTON,PA
ATTENDED BETHLEHEM CENTER
 THIS STUDENT ATTENDED SCHOOL   7 TERMS
 TERM NUMBER   1,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


CHEM 11        4   B
ENG 1          3   B
MATH 21        4   B
POL SC 1       3   C
E G 1          2   D
 GPA FOR TERM   1 IS   2.56
 TERM NUMBER   2,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


E G 2          2   B
MATH 22        4   B
CHEM 12        4   A
ENG 2          3   B
POL SC 2       3   C
 GPA FOR TERM   2 IS   3.06
 TERM NUMBER   3,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


PHIL 1         3   B
M E 22         3   D
PHYS 95        4   C
M E 40         1   B
MATH 23        4   B
 GPA FOR TERM   3 IS   2.33
 TERM NUMBER   4,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


P ED 1         1   B
MATH 24        4   B
M E 50         3   D
T A M 23       3   D
PHYS 96        4   C
 GPA FOR TERM   4 IS   1.93
 TERM NUMBER   5,   2 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


T A M 24       3   B
MATH 110       3   A
 GPA FOR TERM   5 IS   3.50
 TERM NUMBER   6,   6 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ECON 80        3   B
I E 33         3   B
I E 22         3   B
P ED 2         1   A
PHYS 97        4   D
T A M 27       3   B
 GPA FOR TERM   6 IS   2.59
 TERM NUMBER   7,   6 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...
```

SAMPLE OUTPUT

Figure 12 (cont.)

SAMPLE OUTPUT

Figure 12 (cont.)

```
ACCT 77         3   B
I E 121         3   A
I E 171         3   B
ENG 192         2   B
PSYCH 132       2   B
E E 11          3   F
 QPA FOR TERM   7 IS   2.63
 QUALITY POINT AVERAGE IS  2.58
 TOTAL CREDITS = 101.00
 TOTAL QUALITY POINTS =261.00
```

```
WILKINSON, ROBERT E. F-424-489      21 EDGEWOOD DRIVE, NORWALK,OHIO
MALE    DECEMBER 27,1936  SINGLE
ATTENDED OHIO WESLEYAN
 THIS STUDENT ATTENDED SCHOOL   3 TERMS
 TERM NUMBER   1,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I E 43         3   B
MET E 9        3   B
MATH 22        4   B
C G 1          2   B
 GPA FOR TERM   1 IS   3.00
 TERM NUMBER   2,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


PHYS 95        4   C
MATH 23        4   B
I E 22         3   A
I E 23         4   A
 GPA FOR TERM   2 IS   3.20
 TERM NUMBER   3,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


MATH 24        4   A
I E 171        3   B
I A M 23       3   B
PHYS 96        4   C
PSYCH 132      2   C
 GPA FOR TERM   3 IS   2.88
 QUALITY POINT AVERAGE IS  3.02
 TOTAL CREDITS =  43.00
 TOTAL QUALITY POINTS =130.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
LUTHER,RICHARD D.    G-240-320
111 LAFAYETTE AVE.
PITTSBURGH 13, PA.
 THIS STUDENT ATTENDED SCHOOL   1 TERMS
 TERM NUMBER   1,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...

ENG 1          3   C
CHEM 11        4   C
MATH 21        4   A
E G 1          2   B
ECON 80        3   C
 QPA FOR TERM    1 IS   2.63
 QUALITY POINT AVERAGE IS  2.63
 TOTAL CREDITS =  16.00
 TOTAL QUALITY POINTS = 42.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
WILKINSON, ROBERT E. F-424-489     21 EDGEWOOD DRIVE, NORWALK,OHIO
MALE     DECEMBER 27,1936  SINGLE
ATTENDED OHIO WESLEYAN
 THIS STUDENT ATTENDED SCHOOL   4 TERMS
 TERM NUMBER   1,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I E 43          3   B
MET E 9         3   B
MATH 22         4   B
E G 1           2   B
 GPA FOR TERM   1 IS   3.00
 TERM NUMBER   2,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


PHYS 95         4   C
MATH 23         4   B
I E 22          3   A
I E 23          4   A
 GPA FOR TERM   2 IS   3.20
 TERM NUMBER   3,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


MATH 24         4   A
I E 171         3   B
T A M 23        3   B
PHYS 96         4   C
PSYCH 132       2   C
 GPA FOR TERM   3 IS   2.88
 TERM NUMBER   4,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I E 172         3   A
I E 178         3   B
I E 121         3   C
MATH 110        3   A
E G 2           2   B
 GPA FOR TERM   4 IS   3.21
 QUALITY POINT AVERAGE IS  3.07
 TOTAL CREDITS =  57.00
 TOTAL QUALITY POINTS =175.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
LUTHER,RICHARD D.     G-240-32C
111 LAFAYETTE AVE.
PITTSBURGH 13. PA.
 THIS STUDENT ATTENDED SCHOOL   2 TERMS
 TERM NUMBER   1,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ENG 1           3   C
CHEM 11         4   C
MATH 21         4   A
E G 1           2   B
ECON 80         3   C
 GPA FOR TERM   1 IS   2.63
 TERM NUMBER   2,   4 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


CHEM 12         4   C
ENG 2           3   C
E G 2           2   B
MATH 22         4   A
 GPA FOR TERM   2 IS   2.77
 QUALITY POINT AVERAGE IS  2.69
 TOTAL CREDITS =  29.00
 TOTAL QUALITY POINTS = 78.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
RIUNDI,RICHARD F. JR3E-030-524
BOX 356  FRED ERICKTON,PA
ATTENDED BETHLEHEM CENTER
 THIS STUDENT ATTENDED SCHOOL   8 TERMS
 TERM NUMBER   1,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


CHEM 11        4   B
ENG 1          3   B
MATH 21        4   B
POL SC 1       3   C
E G 1          2   D
 GPA FOR TERM   1 IS   2.56
 TERM NUMBER   2,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


E G 2          2   B
MATH 22        4   B
CHEM 12        4   A
ENG 2          3   B
POL SC 2       3   C
 GPA FOR TERM   2 IS   3.06
 TERM NUMBER   3,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


PHIL 1         3   B
M E 22         3   D
PHYS 95        4   C
M E 40         1   B
MATH 23        4   B
 GPA FOR TERM   3 IS   2.33
 TERM NUMBER   4,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


P ED 1         1   B
MATH 24        4   B
M E 50         3   D
I A M 23       3   D
PHYS 96        4   C
 GPA FOR TERM   4 IS   1.93
 TERM NUMBER   5,   2 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I A M 24       3   B
MATH 110       3   A
 GPA FOR TERM   5 IS   3.50
 TERM NUMBER   6,   6 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ECON 80        3   B
I E 33         3   B
I E 22         3   B
P ED 2         1   A
PHYS 97        4   D
I A M 27       3   B
 GPA FOR TERM   6 IS   2.59
 TERM NUMBER   7,   6 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


ACCT 77        3   B
I E 121        3   A
I E 171        3   B
```

SAMPLE OUTPUT

Figure 12 (cont.)

```
[illegible] 192        2   B
PSYCH 132      2   B
E E 11         3   F
 QPA FOR TERM    7 IS    2.63
 TERM NUMBER    8,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...


I E 178        3   C
I E 172        3   B
I E 50         3   D
I E 122        3   A
F E 11         3   B
 QPA FOR TERM    8 IS    2.60
 QUALITY POINT AVERAGE IS  2.59
 TOTAL CREDITS = 116.00
 TOTAL QUALITY POINTS =300.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

SAMPLE OUTPUT

Figure 12 (cont.)

```
PHIL 1      COURSE IS NOT REQUIRED FOR STUDENT  CRAWFORD, RONALD E.  WITH CURRICULUM NO.          2
CRAWFORD, RONALD E. E-078-471      70 EASY STREET
 L 7-2103   UNIONTOWN SENIOR HIGH SCHOOL
DECEMBER 8, 1939
 THIS STUDENT ATTENDED SCHOOL    8 TERMS
 TERM NUMBER    1,    5 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


CHEM 11         4   C
E G 1           2   C
ENG 1           3   B
MATH 21         4   B
POL SC 1        3   C
 GPA FOR TERM    1 IS    2.44
 TERM NUMBER    2,    5 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


ENG 2           3   B
E G 2           2   B
MATH 22         4   A
CHEM 12         4   C
POL SC 2        3   C
 GPA FOR TERM    2 IS    2.81
 TERM NUMBER    3,    4 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


ECON 80         3   B
PHYS 95         4   F
P ED 1          1   A
MATH 23         4   A
 GPA FOR TERM    3 IS    2.42
 TERM NUMBER    4,    3 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


MATH 24         4   A
PHYS 96         4   B
CH E 83         4   B
 GPA FOR TERM    4 IS    3.33
 TERM NUMBER    5,    4 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


T A M 23        3   B
MATH 110        3   A
PHYS 95         4   A
INTDIS 81       3   B
 GPA FOR TERM    5 IS    3.54
 TERM NUMBER    6,    4 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


PHYS 97         4   C
M E 50          3   C
ENG 192         2   A
T A M 27        3   B
 GPA FOR TERM    6 IS    2.58
 TERM NUMBER    7,    4 COURSES WERE TAKEN.     THEY ARE AS FOLLOWS...


I E 33          3   B
ACCT 77         3   A
T A M 24        3   D
I E 171         3   C
 GPA FOR TERM    7 IS    2.5
 TERM NUMBER   8,   5 COURSES WERE TAKEN.    THEY ARE AS FOLLOWS...

P ED 2         1   A
I E 172        3   B
PHIL 1         3   A
I E 22         3   B
I E 42         3   B
 QPA FOR TERM   8 IS   3.31
 QUALITY POINT AVERAGE IS  2.86
 TOTAL CREDITS = 106.00
 TOTAL QUALITY POINTS =303.00
```

SAMPLE OUTPUT

Figure 12 (cont.)

SAMPLE OUTPUT

Figure 12 (cont.)

```
CRAWFORD, RONALD E.  CAN TAKE COURSE  I E 121   HIS CURRICULUM NO IS    2
CRAWFORD, RONALD E.  CAN TAKE COURSE  M E 23    HIS CURRICULUM NO IS    2
CRAWFORD, RONALD E.  CAN TAKE COURSE  PSYCH 132 HIS CURRICULUM NO IS    2
CRAWFORD, RONALD E.  CAN TAKE COURSE  I E 178   HIS CURRICULUM NO IS    2
CRAWFORD, RONALD E.  CAN TAKE COURSE  I E 51    HIS CURRICULUM NO IS    2
CRAWFORD, RONALD E.  CAN TAKE COURSE  E E 11    HIS CURRICULUM NO IS    2
BIONDI,RICHARD F. JR CAN TAKE COURSE  M E 23    HIS CURRICULUM NO IS    2
BIONDI,RICHARD F. JR CAN TAKE COURSE  I E 42    HIS CURRICULUM NO IS    2
BIONDI,RICHARD F. JR CAN TAKE COURSE  I E 51    HIS CURRICULUM NO IS    2
BIONDI,RICHARD F. JR CAN TAKE COURSE  I E 180   HIS CURRICULUM NO IS    2
WILKINSON, ROBERT E. CAN TAKE COURSE  I E 33    HIS CURRICULUM NO IS    1
WILKINSON, ROBERT E. CAN TAKE COURSE  M E 23    HIS CURRICULUM NO IS    1
WILKINSON, ROBERT E. CAN TAKE COURSE  PHYS 97   HIS CURRICULUM NO IS    1
WILKINSON, ROBERT E. CAN TAKE COURSE  T A M 24  HIS CURRICULUM NO IS    1
WILKINSON, ROBERT E. CAN TAKE COURSE  T A M 27  HIS CURRICULUM NO IS    1
LUTHER,RICHARD D.    CAN TAKE COURSE  P ED 2    HIS CURRICULUM NO IS    2
LUTHER,RICHARD D.    CAN TAKE COURSE  I E 22    HIS CURRICULUM NO IS    2
LUTHER,RICHARD D.    CAN TAKE COURSE  MATH 23   HIS CURRICULUM NO IS    2
LUTHER,RICHARD D.    CAN TAKE COURSE  PHYS 95   HIS CURRICULUM NO IS    2
LUTHER,RICHARD D.    CAN TAKE COURSE  I E 121   HIS CURRICULUM NO IS    2
```

BIBLIOGRAPHY - ABSTRACTS

by

PAUL M. DARGAY

Author: Abendroth, W.W.

Association: Case Institute of Technology

Article: How to Make Examinations Fit the Master Class Schedule

Source: College and University Business, May, 1962, pp. 47-51

Computer: Sorting machines

Abstract: Examination schedule cards are processed on a sorter and the load cards generate conflict cards. These conflicts are resolved by a matrix method which can be done in 3 days. Research is being done on "optimal scheduling" even though it required only 6 man days to assign courses to a pre-determined master schedule.

Author: Anderson, G.E.

Association: New England School Development Council

Article: An Algorithm for Assigning Pupils to Classes

Source: Mimeographed Sheets, March 11, 1963, 31 pages

Computer: IBM 7090

Abstract: This is a very fast assignment program to a manually prepared Master Schedule. Assignments are made by creating a "work space" for each student. This is a matrix: rows are a student's course requests and the columns are the possible meeting time periods. Various combinations of sections are attempted if a course cannot be scheduled. Assumes interchangeable sections for a course. If a student cannot be scheduled in a specified number of trials, conflicts are resolved by school administrators.

Program can be used to schedule 2 terms at once.

Algorithm will assign schedules to the students but they may not be the best schedules.

Author: Appleby, S. J., Black, D.V., and Newman, E.A.

Article: Techniques for Producing School Timetables on a Computer and Their Application to Other Scheduling Problems

Source: The Computer Journal, January, 1961, pp. 237-245

Abstract: The first timetable construction consisted of filling in a blank timetable by assigning the most difficult courses using the criterion of "line" fitting (a course at a time level in the week).

The second program dealt with groups of lines but ended with poor distribution over the entire week.

The third program prepared a table of masters against periods for a given class and also produced a table of classes against periods for a given master. This insures enough periods for single-master and single-class groups.

The overall approach consisted of constructing the timetable by fitting a course or group of courses to a timetable with the criteria themselves changing with the situation.

Author: Austin, David B. and Gividen, Noble

Association: Secondary School Administration Series

Book: The High School Principal and Staff Develop the Master Schedule

Source: Bureau of Publications, 1960, Teachers College, Columbia University, New York

Abstract: A high school principal constructs a "manual" master schedule. Most of the up to date scheduling terms and devices were utilized in the formulation of the schedule. The authors advocate the use of a variety of card systems for scheduling and suggests the computer be used for "organization and use of data" and not to control schedule design.

Author: Barnes, W., Boyd, A.G., and Veneziano, N.

Article: Master Scheduling Program for Chicago High Schools

Source: Paper, January, 1962, 26 pages

Abstract: The Master Scheduling program is part of an integrated system and it makes use of master computer files generated in other routines. Master Schedule constructed on individual student's course elections with objective function being minimization of course conflicts. Uses decision loops which are off line decisions that prolong time of computer run.

Program is a three-stage process of 1) Resource Planning 2) Detail Scheduling 3) the Proof Run.

Scheduling preceded by summarization of a Course Conflict Chart and then a sequence of steps for assigning students, teachers, rooms, etc. by priority rankings.

Author: Blakesley, James F.

Association: Purdue University

Article: Registration Is Now a Matter of Minutes

Source: College and University Business, 27, November, 1959, pp. 38-44

Abstract: Purdue's scheduling method is essentially semi-automated with a pre-determined master schedule to which students are assigned. Conflicts are resolved manually after students have been assigned to their sections.

Author: Blakesley, James F.

Association: Purdue University

Article: A Data Processing System for Scheduling University Students to Classes Using an Intermediate Speed Digital Computer

Source: Master's Thesis, Purdue University, June, 1960

Computer: Datatrom 204

Abstract: This thesis describes the data processing system used to schedule students to a pre-determined master schedule. The program automates the assigning of students to classes.

It contains a comprehensive report of all procedures, of all accounting records, and of the machine program flow charts for a "Data Processing" scheduling system.

Author: Brubaker, James E.

Article: Computer Learning Theory in University Scheduling

Source: Master's Thesis, University of Pittsburgh, 1961

Computer: IBM 7070

Abstract: This thesis quickly demonstrates that combinatorial aspects of a scheduling problem prohibit considerin all possible schedules to arrive at an optimal answer in a reasonable length of time. The scheduling problem is given in terms of a model and rules which reduce the number of possible schedules and make criterion testing of the solutions possible.

The scheduling problem is divided into a two-phase problem, master schedule created first with the assignment of students to the master schedule following.

Brubaker's research of his scheduling simulator is detailed chronologically. Machine learning routines are explained and a mathematical discussion of phase 2 is given.

Author: Bush, Robert N.

Association: Stanford University

Article: A New Design for High School Education: Assuming a Flexible Schedule

Source: National Association of Secondary School Principals, May, 1962, pp. 30-34

Abstract: A flexible schedule is possible by use of electronic data processing machines. Professor Oakford of the Stanford Industrial Engineering Department is directing pioneering work in the application of modern procedures to high school scheduling.

Author: Bush, R.N., Caffrey, J., Oakford, R.V., and Allen, D.W.

Article: Using Machines to Make the High School Schedule

Source: School Review, 69, Spring, 1961, pp. 48-59

Abstract: Initially starting with a tentative master schedule Stanford revises it until the desired or final program is completed.

The article also includes a matrix solution of class conflicts.

Author: Bushnell, D.R.

Association: System Development Corporation

Article: The Computer Assisted School System

Source: Computers and Automation, February, 1962, pp. 6-9

Abstract: A teaching system is explained by using teaching machines. The article mentions that they are new techniques for necessary educational data such as real time registration and scheduling programs.

Author: Csima, J. and Gotlieb, C.C.

Association: Institute of Computer Science, University of Toronto

Article: A Computer Method for Constructing School Time Tables

Source: Paper presented to 18th Annual Conference of the Association of Computing Machinery, August, 1963, 17 pages

Computer: IBM 7090

Abstract: A computer method of constructing time tables uses iterations involving boolean matrices. Starting with pre-assignments a search is made for "tight" combinations and the availability matrices are reduced till completion or until it is indicated that no time table is possible with the given pre-assignments. The computer tests used 9 teachers, 9 classes and 9 hours. Each teacher must meet each class once in constructing a feasible time table.

The paper also gives references which relate theorems on matrices of zero and one and on bipartite graphs to time table construction. Authors also state some problems of applying their method in real situations.

Author: Dargay, P.M., Helmer, F.T., Kurki, A.W., and Powell, C.E.

Article: Turtle Creek High School Scheduling Problem

Source: Industrial Engineering Department, University of Pittsburgh

Computer: IBM 7094

Abstract: The procedure used in applying linear programming to the scheduling problem is outlined in the report. The student conflict matrix and pertinent data of teachers, rooms, and policy at Turtle Creek were used in formulating the restrictions. Using the LP/90 program, an optimal solution was found and alternative optimum solutions were indicated.

Author: Donaghey, C.E. and Ozkul, O.S.

Association: Industrial Engineering Department, University of Pittsburgh

Article: A Master Schedule Generator and Schedule Improvement Program

Source: Mimeograph Sheets, February 4, 1963, 11 pages

Computer: IBM 7070

Abstract: A feasible master schedule generated randomly. Room and faculty members assigned to courses. Conflicts are checked in the process of generation of the master schedule. Students are assigned to course sections, and conflicts are counted. The course that causes the most conflicts is rescheduled. Students are assigned to course sections again and the cycle is repeated.

Author: Dwyer, A.J.

Association: Philco Corporation

Article: Automated Teaching Study and Experimentation

Source: Computers and Automation, November, 1961, p. 22

Computer: Philco 2000

Abstract: CLASS is a project being carried on by the Systems Development Corporation of Santa Monica, California. CLASS is Computer Based Laboratory for Automated School Systems. One of the benefits of the project may be that registration scheduling will be done automatically.

Author: Flanagan, Scott

Association: Huntington Beach Union High School, Huntington Beach, California

Article: Machine Programming at Huntington Beach High School

Source: Journal of Secondary Education, 36, October, 1961, pp. 371-373

Abstract: Pre-registration worksheet prepared and tallied before IBM machine develops a tentative conflict schedule. Conflicts set in matrix form and reduced to a minimum. The tentative master schedule is prepared and sorting assigns students to classes.

Author: Green, T.C. and Smith, E.T.

Association: International Business Machines Corporation

Article: Class Loading and Student Scheduling Using 7070 (7074) and 1401

Source: IBM, Cleveland - Version "C", July 15, 1962

Computer: IBM 7070 (7074) and 1401

Abstract: This manual contains description of the scheduling process, input-output formats and operating instructions of CLASS. The Master Schedule is determined by test runs on the original draft. An "Alpha" factor provides better classroom balancing and also permits some flexibility in scheduling a combination of sections to a student. The new method is faster and has much information, giving details.

Author: Harding, Robert E.

Association: Industrial Engineering Department, University of Pittsburgh

Article: Linear Programming as an Aid in Solving Scheduling Problems, A Progress Report

Source: Paper Delivered to AEDS Washington Conference, May 12, 1963

Abstract: The paper 1) briefly defines the scheduling problem; 2) presents a detailed mathematical formulation of the job of generating master time schedules subject to any set of generally used policy, classroom, student and instructional constraints; 3) shows the special advantages of using these formulations. If criteria exist for comparing alternative feasible master schedules the formulation will provide the best one. If there are alternative schedules as good, then this technique will so indicate and provide means for finding these. In addition, educational management will be provided with a precise means for measuring the effect of the various policies which pertain to scheduling. If inconsistent policies exist, the model indicates it. If no criteria have been established, the model will generate feasible master schedules if there are any (schedules which violate no constraints); 4) outlines plans for further research and for utilizing the problem formulation presented and 5) works a sample problem of small size.

Author: Harding, Robert E.

Association: Industrial Engineering Department, University of Pittsburgh

Article: The Educational Scheduling Problem

Source: Mimeographed sheets, University of Pittsburgh, 1963

Abstract: The author defines the general "educational scheduling problem" facing all types of educational institutions in terms of a number of sub-problems; 1) a determination of those courses which will be offered, 2) which time periods are available for scheduling, 3) assignment of course offerings to time periods, 4) assignment of course offerings to classrooms, 5) determination of student demand, 6) assignment of instructors to courses, and 7) assignment of instructors to course offerings. He also discusses the goals of educational organizations and criteria for educational scheduling, the constraints which reduce the size of the set of alternative courses of action in scheduling, and areas of needed research. Two of the items indicated as part of the latter discussion are 1) quantification techniques for criteria in educational scheduling - the development of practical, operational methods which are realistic and 2) the design of methods for measuring and using student demand in educational scheduling.

Author: Holz, Robert E.

Association: Massachusetts Institute of Technology

Article: Generalized Academic Simulation Programs

Source: GASP Reports, Massachusetts Institute of Technology

Computer: IBM 709/7090

Abstract: MIT divides the scheduling problem into 2 parts: 1) Timetable construction - includes assignment of time patterns, instructors, and classrooms to classes, and 2) sectioning - student assignment to classes.

A master schedule is generated by the timetable construction and then students are assigned to classes by presenting their course requests in a matrix format which optimizes scanning and analyzing these requests. The master schedule is generated by teacher vectors and a sample of students tested against the times constructed.

Author: Holzman, A.G., and Turksen, I.B.

Association: Industrial Engineering Department, University of Pittsburgh

Article: Boolean Isomorphic Structures in an Educational Scheduling Model

Source: Monograph, February 4, 1963

Computer: IBM 7070

Abstract: A general and amorphous model of the educational scheduling system is proposed. Student demand, faculty resources, administrative regulations, and physical plant facilities are described. Mathematical algorithms are developed to determine the feasible solution space for a master schedule. Results of a 10-variable problem are given and conclusions are drawn.

Author: Howell, Floyd, Jr.

Association: University of Miami

Article: University to Use Computer for Paperwork, Medical and Fisheries Data

Source: Franklin Institute Journal, 272, July, 1961, pp. 83-84

Computer: RCA 301

Abstract: Machine will be used for registration at the University of Miami and for other functions such as medical data processing and fisheries data.

Author: Kiviat, Philip

Association: Industrial Engineering Department, University of Pittsburgh

Article: Optimal Scheduling in Educational Institutions: Progress Report for National Science Foundation

Source: Typed sheets, University of Pittsburgh, 13 pages

Abstract: Procedure is one of the aggregation of student course demand and the creation of a master schedule which is a probabilistic reflection cf individual student demand. The procedure generates schedules based on a-priori estimates of student demand. It then assigns students by means of an iterative algorithm which converges to a best overall student assignment.

The method assigns groups of courses to available time periods using a probabilistic measure of student course selection. By partitioning the total courses offered into schedule groups and then using the boolean vectors of these group requests, a student schedule group conflict pattern matrix is constructed. The schedule groups are then assigned to time periods in a heuristic manner until a Master Schedule is generated. This Master Schedule can then be refined by further partitioning of the groups and assigning them to smaller time units.

Author: Kornfeld, Leo

Article: What Every Schoolman Should Know about Data Processing

Source: School Management, October, 1962, pp. 59-74

Abstract: The uses are 1) grade reporting, 2) attendance records, 3) bookkeeping, 4) library records, 5) contracted services which include scheduling.

Author: Kornfeld, Leo

Article: What Every Schoolman Should Know About Data Processin

Source: School Management, 11, November, 1962, pp. 88-90

Computer: IBM 7074

Abstract: Case #4 - A Case Study of Contracted Services

A service bureau does the scheduling for a Long Island, N.Y. school district at a cost of $1 per student.

Pre-registration and a preliminary master schedule are done before a trial run on the computer.

Corrected master schedule reduces the conflict to a minimum. Final conflicts resolved by counselors.

Author: Merz, Albert F.

Association: Nyack High School, Nyack, New York

Article: The Data Processing Equipment for Educational Records

Source: National Association of Secondary School Principals, April, 1962, pp. 8-12

Abstract: Data processing is used for various administrative and clerical functions in the educational system. It can also be used for scheduling.

The method of scheduling consists of pre-registration with a tally of courses after the requests have been sorted. Teachers are also assigned but conflicts are resolved manually.

Author: Oakford, Robert V.

Association: Industrial Engineering Department, Stanford University

Article: Machine Assistance for Constructing the High School Schedule: An Industrial Engineer's Report

Source: Journal of Secondary Education, October, 1961 pp. 374-379

Abstract: Scheduling problem is divided into logical decisions handled by the computer and judgment decisions handled by the high school principal. The first step is to construct a timetable for class meetings and then assign students to these classes, or a combination of the two steps to solve the scheduling project.

Project is under a grant from the Ford Foundation with Stanford using a one phase approach with provisions for resolution of apparent conflicts.

Author: Pack, Charles R.

Association: Stanford University

Article: Mechanization of a Variable High School Schedule

Source: Thesis, Stanford University, June, 1961

Computer: Burroughs 220

Abstract: The Master Schedule is generated by matching teacher vectors with room vectors for each course offered. The students who want the particular course are then scheduled at this time if they are free. This conflict can be resolved by re-scheduling the time of teachers - room or schedule free students to various sections. All teachers, rooms and student course requests are represented in Boolean-vector form. Boolean manipulations are used in the assigning of the students and are a very important factor in conflict resolution. Test run on 100 students with various programs (courses) proved that a master schedule could be generated but this generating should be handled by a fast computer.

Author: Sabbagh, Neva

Association: Purdue University

Article: University Student Scheduling by Datatron

Source: Computers and Automation, March, 1959, p. 13

Abstract: Scheduling is done by an electronic data processing system. The scheduling process is an assignment procedure. Students are assigned to a predetermined master schedule. Conflicts are resolved manually.

Author: Sherman, G.A.

Article: Combinatorial Scheduling: On Finding A Partition of a Finite Set Which Maximizes a Set Function

Source: Doctoral Thesis, Purdue University, 1961

Abstract: This is a detailed mathematical thesis which formulates the scheduling problem into a problem of finding an optimal partition of a finite set of points with the criterion being the minimization of the expected number of conflicts for a randomly drawn student.

Two classes of convex polyhedra emerge when the problem is related to integer linear programming problems. The two classes are difficult to use in linear programming algorithms, so the author develops a stochastic type algorithm to find optimal or near optimal partitions.

Author: Skillings, H. H.

Article: Manual for Computer Class Scheduling: Mechanics of Operating at the University of Massachusetts

Source: Office of Institutional Studies, University of Massachusetts, April, 1963

Computer: IBM 7090

Abstract: The first step is a preparation of a master schedule of courses and sections being offered. Pre-registration is completed by each student against this master schedule. The student requests are then assembled into student decks which are sorted and compiled into summary of course enrollments and possible adjustments of the master schedule. The students are assigned to classes by the computer to the adjusted master schedule. Conflicts are manually resolved.

Author: Turksen, I.B.

Association: Industrial Engineering Department, University of Pittsburgh

Article: Boolean Transformation in a Scheduling Model

Source: Master's Thesis, University of Pittsburgh, June, 1962, 84 pages

Computer: IBM 7070

Abstract: An educational scheduling system is expressed as a model. Inputs to the system include student demand, faculty availability, administrative rules, and physical plant resources. Algorithms are developed for determining courses to be offered, number of sections of each, and construction of a feasible solution space of the master schedule. A small scale example is analyzed by the structures developed.

Author: Welton, Greg

Association: Burroughs Corporation

Article: Data Processing and the School Schedule, a Burroughs Approach

Source: Journal of Secondary Education, October, 1961, pp. 382-384

Abstract: Scheduling is a two-phase operation. The first phase consists of preparing or constructing an overall academic schedule of classes. The second phase is the scheduling of the students to this pre-determined master schedule.

Author: Wulff, B.H.

Association: International Business Machines Corporation

Article: Data Processing for Student Scheduling: An IBM Approach

Source: Journal of Secondary Education, October, 1961, pp. 380-381

Abstract: Electronic data processing method of scheduling using a Student Program Card (request) and a Master Class Card (schedule results).

Author: Yasaki, Ed

Article: Educational Data Processing

Source: Datamation, 9:6, June, 1963, pp. 24-27

Abstract: The Workshop on Educational Data Processing covered six topic areas: scheduling, system design and analysis, information retrieval systems, simulation and modeling, school administrative services, and programming scheduling.

Purdue and the University of Pittsburgh are trying to construct an "optimum" schedule of classes. Dr. Holzman and Mr. Turksen of Pitt are researching the feasibility of generating a master schedule.

Author: Zionts, Stanley

Association: Industrial Engineering Department, University of Pittsburgh

Article: A Method of Partitioning Student Course Requirements for Probabilistic Assignment to Time Schedules

Source: Typed Sheets, University of Pittsburgh, 9 pages

Abstract: The author uses the principle that the object of partitioning is to maximize the probability that a student taking any one course in a partitioned group of courses is taking all the courses in the group. Therefore, only one partitioning operation is required for this high interaction.

This is set up as a linear programming problem - maximize the interaction probabilities. The problem is also set up in a Transportation Structure with the diagonal elements assigned (-M) values. When an optimum is reached, the solution on either side of the diagonal can be used as a partition. Once the partition is complete, the Kiviat algorithm is used to schedule the group relative to one another in different time portions of the schedule week.

Article: Data Processing for Secondary Schools

Source: Customer Instruction Manual, Service Bureau Corporation, 25 pages

Abstract: The booklet is an instruction manual that details the procedure high schools must follow to have their schools scheduled by the Service Bureau Corporation.

From student course listings, a tally is prepared which is used as an aid in a manual preparation of a master schedule. Students are assigned to classes and student reports of assigned and unassigned requests are listed. Using a population report sheet which lists number of seats available in each course, the school administrator resolves the conflicts.

Article: High School Scheduling Done by IBM Machines

Source: Personnel and Guidance Journal, May, 1957, p. 568

Abstract: Assignment of students and teachers to a pre-determined master schedule before master schedules are duplicated on detail cards. The student cards are sorted to find class conflicts with conflicts resolved after scheduling is finished.

Article: Points of Interest

Source: Data Processing Digest, April, 1962, p. 20

Abstract: Student registration and pupil accounting are being handled by the Univac Optical Scanning Punch at the Board of Education, Baltimore County, Maryland

---

Article: Rochester University to Assist High Schools in Class Scheduling

Source: Date Processing for Education, Vol. 1, No. 1, November, 1962, p. 10

Abstract: 2800 pupils of Rochester East High were scheduled via computer in 17 minutes compared to the 480 man-hours usually required in a large size school. Included were 2800 weekly class schedules. 550 class lists, one for each teacher, a master schedule of classes and enrollment for the entire school.

The scheduling conflicts were 1/2 of 1% compared to the customary 5% conflicts.

Article: West Virginia to Offer Scheduling Package at Low Cost to Local Schools

Source: Association for Educational Data Systems Bulletin, May, 1963, p. 9

Computer: IBM 1620

Abstract: 15 West Virginia schools will schedule their students using a package program offered by the State Department of Education and the University of West Virginia. Estimated scheduling price will be approximately 25 cents per pupil.